A FRAMEWORK FOR KNOWLEDGE DISCOVERY FROM SPARSE, HIGH-DIMENSIONAL MEDICAL DATASETS

BY

CHANDRASEKAR RAMACHANDRAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2009

Urbana, Illinois

Adviser:

Professor Jiawei Han

Abstract

In this work, we describe a comprehensive framework for knowledge discovery from medical records called *SDM-Miner*. The records are created before, during and after pancreatic islet cell transplantation1 on a group of diabetic patients. The knowledge discovery focuses on selecting the most relevant variables for predicting the outcome of islet cell transplants temporally, and supporting the medical understanding of the variable relationships that would lead to insulin-free outcome of a transplant with machine learning models. The challenges of knowledge discovery lie in the temporally sparse nature of medical records and the large number of variables which make the traditional statistical analyses ineffective. Our approach to overcome the challenges is to combine data-driven computationally intensive modeling with statistical modeling. The framework incorporates this approach during three phases of knowledge discovery including (1) statistical data-preprocessing, (2) pattern search based dimensionality reduction, and (3) association rule based and conditional probability based data-driven modeling.

We evaluate the framework by cross validating the models (of machine learning) using prediction errors and uncertainty of rule discovery. In order to demonstrate the novelty of the framework and the improved performance in knowledge discovery, we report results using real and synthetic datasets. Experimental results on synthetic data act as a sanity check in order to verify the effectiveness of our models in the absence of standard test results. The evaluation results show that our framework led to smaller mean error with the decreasing number of variable samples, higher robustness to Gaussian noise, and higher confidence and support of association rules than the previous methods. Furthermore, we evaluate our proposed technique using existing machine learning algorithms such using the Weka toolkit and show the improved performance of our work as compared to previous approaches.

To My Parents.

Acknowledgments

I would like to gratefully acknowledge the enthusiastic supervision of my thesis advisor, Prof.Jiawei Han during this work. I thank Dr.Peter Bajcsy for his wonderful support and guidance as my research supervisor and for the insightful discussions which helped shape my thesis. I would like to thank Luigi Marini for his help in analysing the Islet Cell dataset and for his suggestions on the various data-mining issues that I could address as part of my thesis. This research was supported by National Center for Supercomputing Applications (NCSA), the Image Spatial Data Analysis (ISDA) group. I would also like to acknowledge Dr. Jose Oberholzer, the Director of the Islet and Pancreas Transplant Program at the University of Illinois at Chicago (UIC), for providing the measured data and insights about the challenges. I would also like to thank the administrative staff in the Department of Computer Science for proof-reading the thesis and ensuring that it is submitted on time. And finally, thanks to my parents and numerous friends who endured this long process with me always offering support and love.

Table of Contents

List of Tables . . . . . **vii**

List of Figures . . . . . **viii**

List of Abbreviations . . . . . **ix**

List of Symbols . . . . . **x**

Chapter 1 Introduction **1**
1.0.1 Overview of Medical Data Mining 1
1.0.2 Problem Overview 3

Chapter 2 Background **6**
2.1 Statistical Techniques 6
2.2 Data Mining Techniques 7
2.2.1 Clinical Trials 7
2.2.2 Rule-Mining and Classification Systems 7
2.2.3 Fuzzy Techniques 8
2.3 Applications 8
2.3.1 Medical Prognosis and Survivability Analysis 8
2.3.2 Decision Support Systems 9

Chapter 3 Challenges and Approaches **10**
3.1 Data Representation 10
3.2 Quantifying Algorithm Performance 11
3.2.1 Metric Selection 11
3.2.2 Metric Value Interpretation 11
3.3 Complex Multi-Attribute Relationships 12

Chapter 4 Contributions **17**
4.1 Data Pre-Processing 18
4.2 Dimensionality Reduction 18
4.3 Prediction 20
4.4 Variable Relationships 20

Chapter 5 Theoretical Framework **22**
5.1 Record Structure 23
5.2 Preliminaries 23
5.3 Function-Monotonicity and Rule Measures 26
5.4 Bayes Theorem and Data Partitioning 27

Chapter 6 Data Preprocessing 29
6.1 Overview of Current Techniques 29
6.2 SDM-Estimate Algorithm 32
6.3 Summary 35

Chapter 7 Dimensionality Reduction 36
7.1 Overview 36
7.2 Problem Statement 37
7.3 SDM-Reduction Algorithm 37
7.4 Summary 42

Chapter 8 Rule-Mining and Prediction 43
8.1 Generating Classification-Association Rules 43
8.1.1 Overview 43
8.1.2 Algorithm 44
8.2 Bayesian Prediction 46
8.3 Summary 46

Chapter 9 Experimental Results 47
9.1 Data 47
9.2 Evaluation Metrics and Results 47

Chapter 10 Conclusions and Future Work 52

References 53

List of Tables

3.1 Islet cell transplant dataset with Insulin measurements 14
3.2 Islet cell transplant dataset with Glycemic Index and Insulin measurements . . . 14
3.3 Islet cell transplant dataset with Glycemic Index, Insulin measurements 16

9.1 Results of association rule discovery 48

List of Figures

1.1 Process model for a typical medical data-mining algorithm. 2
1.2 High-level workflow diagram of SDM-Miner. 4

3.1 2-dimensional data. 13
3.2 3-dimensional data. 15
3.3 Visualization using Kohonen's Self Organizing Maps. 16

4.1 Contributions in this work. 17
4.2 A high-level overview of the prediction algorithm. 19
4.3 Steps involved in a generating classification-association rules. 21

5.1 Structure of a typical record. 22

6.1 List of missing values estimation methods. 29
6.2 List of models built based on missing value estimation methods. 30
6.3 The hashed table of missing values. 31
6.4 *SDM-Estimate* hashing algorithm 31
6.5 *LOCATE* algorithm 33
6.6 *SDM-Estimate* algorithm for estimating temporally sparse values 34

7.1 The multi-layered index structure. 39
7.2 Nearest search algorithm 40
7.3 Partitioning algorithm 41
7.4 Partitioning and search algorithm using $m - NS$ data structure 42

8.1 *SDM-Rules* algorithm for discovering classification association rules. 45

9.1 Mean Error(%) Vs Number of Samples for Baseline Synthetic Case. 48
9.2 Mean Error(%) Vs Number of Samples for Random subsampling. 49
9.3 Mean Error(%) Vs Number of Samples for Various σ of Gaussian noise. 50
9.4 Mean Error(%) Vs Percentage of Samples considered as Prior Knowledge for Real dataset. 51

List of Abbreviations

ANN	Artificial Neural Networks.
SDM-Estimate	Algorithm For Estimating Temporally Missing Values
SDM-Rules	Algorithms for Determining Association Classification Rules
DR	Dimensionality Reduction
SDM-Reduction	Dimensionality Reduction Algorithm
CDW	Clinical Data Warehouse
MAR	Missing at Random
MI	Multiple Imputation
CAR	Classification-Association Rules
LOCATE	Algorithm to Locate Missing Value

List of Symbols

D_i	i-Incomplete Set Of Data Points
S	n-Dimensional Space
D_k	k-Complete Set Of Data Points
μ_1	Correlation Estimation Function
μ_2	Residual Estimate
v	Row Vector of D_i
η	Normal Residual
k	Itemset Size
$P(c_k)$	Class Probability
$x(t)$	Sinusoidal Signal Function
N	Total Number of Items in an Instance
ω_s	Base Sampling Frequency
R_i	Record Day
P_i	Patient ID
C	Covariance Matrix
$Err(D)$	Error in Similarity Measures
$M(R_{d_i})$	Minima of Mean Squared Error Function
$H(d_i, d_j)$	Hamming Distance between row vectors d_i and d_j
S_t	Residual at t-th Step
z	A Single Element from a RuleSet
c	Confidence of a Rule
κ	Prediction Error
H	Hash Table Index
τ	Least Squares Regression model Parameter Estimate
γ	Estimate for Missing Value

Chapter 1

Introduction

1.0.1 Overview of Medical Data Mining

Data mining of medical records has grown into a challenging and interesting research area over the past several years[1, 2, 3]. Medical records have been known for the diversity of measurements ranging from auditory and visual sensations to complex recollections of traumas and stress [4]. The ability to store and access data in electronic form has led to a growth of medical records over the years. Notwithstanding the ethical and legal issues involved in medical data access and analysis, the knowledge discovery from medical data has been challenging due to the sheer volume, distributed repositories of data, heterogeneity of variables, naming and unit conventions, temporally sparse nature of variables and a large number of variables, just to name a few. Medical data-mining has also become commercially viable over the past few years [5] and has impacted the growth and development of the healthcare industry in a positive manner. Several different areas of medical diagnosis, treatment and clinical research now depend on data-mining analysis as a core component of their tools.

In our work we deal with medical records obtained from patients who are afflicted with Type-1 diabetes [6], a condition in which the human pancreas produces little or no insulin allowing sugar to enter the body cells. These patients have been given pancreatic islet cell transplants as a treatment to regulate insulin production [6, 7]. Transplantation becomes essential to avoid several complications arising from side-effects of diabetes over a long period of time. However, not all patients are subject to this procedure and only the most severe cases (when whole-organ transplantation has failed) are considered for islet cell transplants [8]. Our goal is to select the most relevant variables for predicting the outcome of islet cell transplants, and support the medical understanding of the variable relationships that would lead to insulin-free outcome of a trans-

plant with machine learning models. This would mean a flow-like data mining framework which would pre-process the existing transplant data (to help in further data mining analysis), perform dimensionality reduction (to choose the most relevant variables), determine the predicted values of different variables and finally determine association rules which would model the interactions between different variables and their relationship to the insulin-free outcome on the patients.

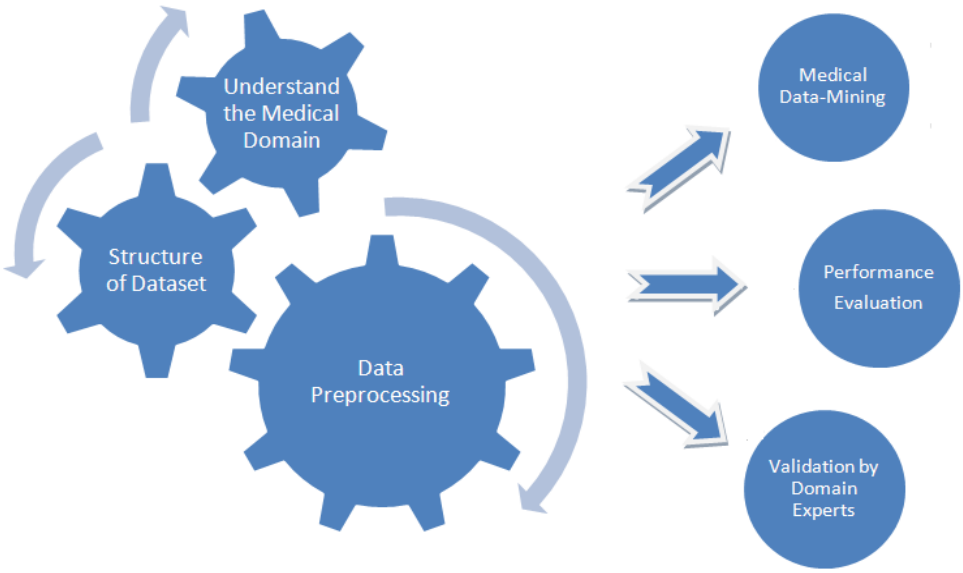

Figure 1.1: Process model for a typical medical data-mining algorithm.

Our task is made difficult by the fact that the outcome of these transplants on the patients is difficult to determine due to (a) the complexity of the biological models and their interactions during such a procedure, (b) the lack of understanding of this complex procedure, (c) the large number of variables involved in a successful transplant procedure, and (d) the high-dimensional[1] and sparse nature of observable measurements involved. The data-mining algorithms developed in this work depend equally on the sparsity and high-dimensionality of the data as they depend

[1]In our work, we use dimensions, attributes and variables interchangeably. Similarly, the medical records pertaining to one patient are denoted as instances, samples or records.

on standard relationship indicators amongst variables present in the dataset. Figure 1.1 shows a typical process model for a medical data mining algorithm that we use in our work. From a clinical perspective there is still a debate as to what variables represent a successful islet cell transplant. Currently it is agreed upon that insulin reduction is one of the key variables.

We also focus on giving a reasonable degree of user interaction in the toolkit that we have developed for this work. Our argument is that since the boundaries of acceptable insulin-free outcome indicators are loosely defined, domain experts are allowed to tune the performance parameters until an acceptable outcome is obtained. Our particular instance of medical data mining can be considered as an instance of some of the constraints imposed on data-mining algorithms which rely on purely automated techniques [9]. One of the major goals is described to keep the user (in our case, the domain expert) informed about the meaning of various parameters and results as often as needed in order to reduce the gap between the analysis and interpretability of results as much as possible.

This raises the question of: Should we use traditional statistical analysis for such datasets or should we follow more advanced machine learning and data mining technqiues? To answer this question, it is worth mentioning about a comparative study between traditional statistical analysis and applications of data mining for medical datasets first done in [50]. Here the prevalence of asthma and other chronic respiratory diseases was studied in around 16,957 Australian children. Here they compared their results with those obtained from a government study in [52]. In the original government study only $14 - 16\%$ of the children were found to be affected whereas it was 27% according to [50]. By using Kohonen's Self Organizing Maps [12], the authors showed a high degree of correlation between nightcough and sleep disturbance. This study showed the inherent advantages of using data-mining and machine learning techniques as compared to pure statistical analysis.

1.0.2 Problem Overview

In this work we motivate the need for integrating statistical and machine learning approaches to solve problems of these types in data-mining. To further examine the need for integrating statistical and machine learning approaches, we need to examine the commonly held viewpoints regarding statistics and machine learning based learning approaches. Traditionally, in statistics,

it is regarded that based on the applicability of a set of models to the problem being solved, a particular model is chosen as a prototype. After a model has been selected, its parameters are estimated [10]. The model is then refined based on these parameters. This methodology has proved to be quite reliable and effective for data which is transparent and easily understood. These techniques do not yield desirable results when the available data has been generated by a complicated process (like the measurements used after islet cell transplants) and thus motivate the need for other approaches.

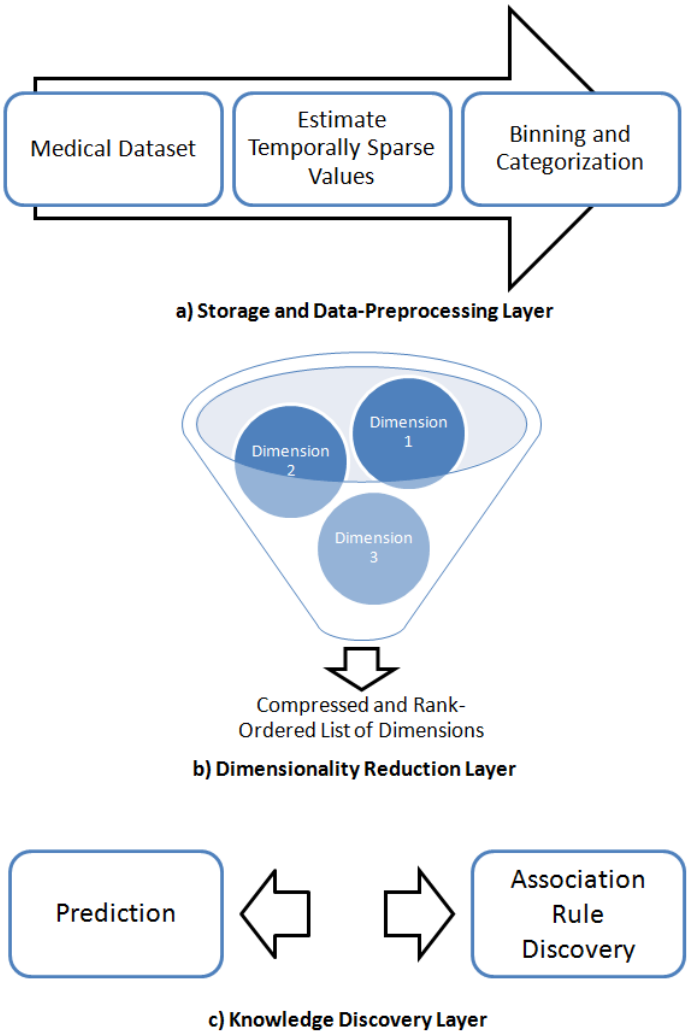

Figure 1.2: High-level workflow diagram of SDM-Miner.

Our main objective is to design a framework for knowledge discovery from medical records that are characterized by sparse and high-dimensional variables. Our approach is depicted in a high-level workflow diagram of SDM-Miner shown in Figure 1.2. This thesis will address the following components of the high-level workflow diagram:

- Pre-processing: Estimate temporally sparse values and categorize variables by user-driven or statistics-driven binning (label assignment).
- Dimensionality Reduction: Determine an error-bounded and rank-ordered list of dimensions which are the most representative of the dataset.
- Temporal Prediction: Predict future attribute values in the dataset with a reasonably high accuracy.
- Variable Relationships: Construct a set of association rules that establishes relationships among various dimensions.

The rest of the work is organized as follows: Some background and related work is given in Chapter 4. Chapter 5 and 6 give an overview of challenges in dealing with sparse and high dimensional variables and a summary of the contributions in this work. Chapter 7 gives a Formal representation of the algorithms used in SDM-Miner and Chapter 8 gives a system-level overview of our work. Furthermore, Chapter 9 and Chapter 10 describe data-preprocessing and dimensionality reduction process respectively. Chapter 10 gives an overview of the algorithms used in knowledge discovery. Chapter 11 presents the experimental results on real islet-cell transplant data as well as some synthetic datasets. Finally Chapter 12 discusses our results and concludes the work while outlining scope for future research.

Chapter 2

Background

In this chapter, we will give a brief overview of some background and related work in medical data mining. There have been several applications of data-mining and knowledge discovery techniques in the medical literature. Since the data is structurally unique, several constraints are imposed on algorithms which analyse this data, and many early techniques which relied on pure statistics or machine learning have not focused on these constraints in much detail. However a detailed overview of the pre-existing techniques in this domain is significant because it helps us understand the growth and development of research in this field.

2.1 Statistical Techniques

Previous work in this domain has focused on specific diseases or entities, for example, coronary artery bypass graft. Analysis of large health databases has been performed to generate statistical models that directly predict outcomes of clinical treatment [18]. Here the authors argue that a combinatorial explosion in terms of number of computations and number of variables under consideration can occur under cases when non-hypothesis driven approaches are considered for large-scale medical data analysis. Unlike earlier techniques which relied on unstructured data collections, recent work has focused on using the information content of large scale patient databases housed at medical institutions. A large proportion of analysis on clinical trials are focused on specific diseases or drugs. For example, caesarean delivery rates [19] examined the treatment procedures impacting the delivery rate on over 250,000 women.

2.2 Data Mining Techniques

2.2.1 Clinical Trials

Recent research [20] has focused on using knowledge discovery and data-mining tools to resolve some of the problems faced by using traditional statistical tools. In general, data-mining techniques such as classifiers, decision trees and regression analysis [12] have been used on medical datasets. Bayesian networks [21], in particular, have been used extensively for representing probabilistic knowledge as a method for pattern recognition in medical datasets. Building on the expected information theory, Robson [3] developed a *Zeta theory* which applied Bayesian networks for a more generalized expected information theory. Association rules were found in hospital infection control and public surveillance data by Brossette et al [22]. Chronic Hepatitis data has been used for mining sequence patterns in [23]. Yin et al. [1] proposed a new concept of direction-setting rules due to the large number of insignificant association rules generated by previous methods. Some other data-mining techniques that have been used exclusively on clinical trial data have incorporated new methods such as false discovery rate calculations by Harrison [24]. This has been necessary because of the difficulties posed in discovering correlated and coincidental patterns in high-dimensional datasets.

2.2.2 Rule-Mining and Classification Systems

Pattern classification systems designed for the measurement and evaluation of various bioprosthetic valves was a subject of study in [27]. This work utilized features computed from spectra of heart sounds. Their study evaluated different types of features and training samples and established that the accuracy of the system was dependent on these two factors. In [28] patterns of interest were discovered in a limited-sized mammographic database by using association rule-mining techniques. In [29] researchers at the University of Calgary determined developed feature classification for mammograms. Techniques for using clinical data sets to perform intelligent temporal rule mining was performed by Khanna et al [44]. Other similar rule mining approaches have focused on epidemiological information systems, computer-aided medical diagnosis and so on. Many of the approaches focus on the reduction in the generation of a large number of rules.

2.2.3 Fuzzy Techniques

A fuzzy approach to medical diagnosis was proposed by John et al [45] and use a concept called as "Fuzzy Cognitive Maps" for their diagnostic tasks. In [46] a novel technique of developing an artificial immune recognition system (AIRS) applied to ECG arrhythmia was proposed. They developed a technique based on fuzzy weighted pre-processing.

Based on our knowledge, the existing systems and past work have yet to address the two fundamental problems posed in our work in detail. First is how to perform knowledge discovery from sparse and high-dimensional data sets? Second, what is the optimal configuration of statistical, pattern recognition and data-mining methods to extract knowledge from medical records with the highest confidence?

2.3 Applications

2.3.1 Medical Prognosis and Survivability Analysis

Medical prognosis and survivability analysis [14] have been prominently studied in the literature. Survivability analysis is a statistical technique to model the time to failure or of an event to occur. There are two different kinds of patients who are modeled using survivability analysis: censored (those who outlast the duration of patient study) and uncensored (those who die before the study terminates). The most common techniques for use in survivability analysis are the Kaplan-Meier method and regression models such as the Cox Proportional Hazard [15]. Burke et al. [16] compared the 5-year predictive accuracy of various statistical models with artificial neural networks (ANNs). This study used the Patient Care Evaluation dataset collected by the Commission on Cancer of the American College of Surgeons. In a recent study, 5-year, 10-year and 15-year breast cancer survivability was predicted by using artificial neural networks and logistic regression models. Other related work in this domain include the work done by Santos-Garcia et al. [17] to estimate cardio-respiratory morbidity.

2.3.2 Decision Support Systems

The knowledge discovery systems have been developed primarily for supporting decision making, for instance, the MediMap project [25]. MediMap utilizes data-mining and decision-support to improve healthcare knowledge management. The project was designed for community health care management. This was a two-phase project where both data-mining and decision support was used to plan the development of public health services. Other related work in this category include clinical data warehouses (CDW), which are used for complex data computations. One example of a CDW is a system built by [26] for treatment of diabetes. Data mining techniques were also used to mine adverse-event databases [42] specifically monitoring the effects of adverse-drug reactions. Study of knowledge discovery from a Veterans Administration Healthcare Information System was explored by Kraft et al [43].

Chapter 3

Challenges and Approaches

This thesis will address four components of the workflow. All components have to overcome basic challenges related to (a)Data Representation, (b) Quantifying Algorithm Performance and (d) Complex Multi-Attribute Relationships. One of the key challenges lies in the fact that the traditional statistical analyses [12] are ineffective for sparse and high-dimensional variables due to several reasons. First, statistical techniques cannot model well non-linear relationships among dimensions. In statistics, there are several theories which model the linear relationships among different variables in a population. One example would be to calculate the mean of a population. On the contrary, in medical data clear linear relationships are absent and thus non-linear hypothesis relationships need to be obtained. For example finding out the age, weight and body height for high cancer risks amongst patients. Popular techniques in medical data mining which use non-linear hypothesis testing include neural networks, Kohonen self-organizing maps and so on [13]. Second, sparse sets of samples of a noisy variable do not provide statistically reliable estimates for analysis. The sparse nature of the data set also makes it difficult to fit a particular model to the dataset. Finally, establishing relevance or irrelevance of a variable using statistical techniques is difficult when dealing with sparse and high-dimensional variables.

3.1 Data Representation

Another challenge which is a common feature of of medical datasets is their heterogeneity [4]. Some of the aspects which contribute to the heterogeneity of data include the different forms of medical data representation. For example, data obtained from medical imaging is difficult to process. Other factors contributing to the challenges in processing medical data include their sheer volume and lack of proper mathematical characterization. There is also no standard form

of representation of medical data and various metrics and measures may not be characterized in a uniform manner. Sometimes patient records are private and are not available to everyone by default. A privacy-preserving approach to mining these patient records needs to be followed to prevent ethical and legal issues from being raised as a result of this data analysis.

In addition to these issues, medical data mining is a challenging process because of the following characteristics of medical data:

1. Organization of data
2. Different units and terminology
3. Naming conventions (like date/time)
4. Different protocols (procedures) to generate values for a variable
5. Granularity of information
6. Uncertainty of variable measurements
7. Policies for editing data

3.2 Quantifying Algorithm Performance

3.2.1 Metric Selection

Another challenge arises when modeling accuracy of variable predictions has to be quantified. Some key metrics for measuring the preciseness of a medical diagnosis (like the impact of islet cell transplants) are sensitivity, specificity and accuracy of test results [4].

3.2.2 Metric Value Interpretation

However, the metrics reporting modeling accuracy could also be misleading. For example, one would generate a set of association rules [12] and all three metrics would have low values because of the sparse and noisy nature of a dataset. There would be instances when a test is positive but it is not detected as such due to the skewed, irregular and sparse distribution of the dataset, and vice-versa. To address this challenge, there is a need to review the quality and sanity of

association rules generated. Clearly algorithms which generate a large number of association rules involving many dimensions are not useful since they are difficult to interpret and to be validated by domain experts like doctors.

3.3 Complex Multi-Attribute Relationships

Yet another challenge is that instances of diseases are affected by the order in which they are linked to other episodes or events. This is called as "episodic data mining" [30]. The use of techniques which do not take these factors into account is not recommended as they do give an accurate model of patient health. Among several data-mining techniques mentioned as useful for medical knowledge discovery include anomaly detection [31], difference detection and so on.

In most medical datasets, relationships amongst attributes are complex and sometimes no clear pattern can be obtained by just observing the datasets. The attributes sometimes have multi-level concept hierarchies which require the expansion and application of the attribute domain at each level. Some related work in this domain focused on deriving multi-level association rules [32]. We also found that the patient datasets obtained for our experimentation are not standardized in organizations, and records are generally entered ad-hoc by nurses and doctors. As a result of this, many patient records are empty. Patient measurements are not taken at the same time of the day for all patients (or in the same day and in the same sequence). This poses additional challenges in integration and analysis.

Now we give examples to show some of the important problems present in analysing sparse, high-dimensional medical data. The first problem is in handling the curse of dimensionality and the second is in the application of some standard machine learning techniques to discover knowledge from this data.

For privacy and security reasons, we will not publish the contents of the islet cell transplant dataset, but rather focus on a small subset which would be enough to demonstrate this example. Table 3.1 shows a small islet cell transplant dataset consisting of insulin measurements at different days for 2 patients. Figure 3.1 gives a visual representation of this data packed into 9 bins. Table 3.2 extends Table 3.1 by showing measurements of Glycemic Index in addition to insulin measurements. Figure 3.2 gives a visual representation of this data packed into 36 bins. Note

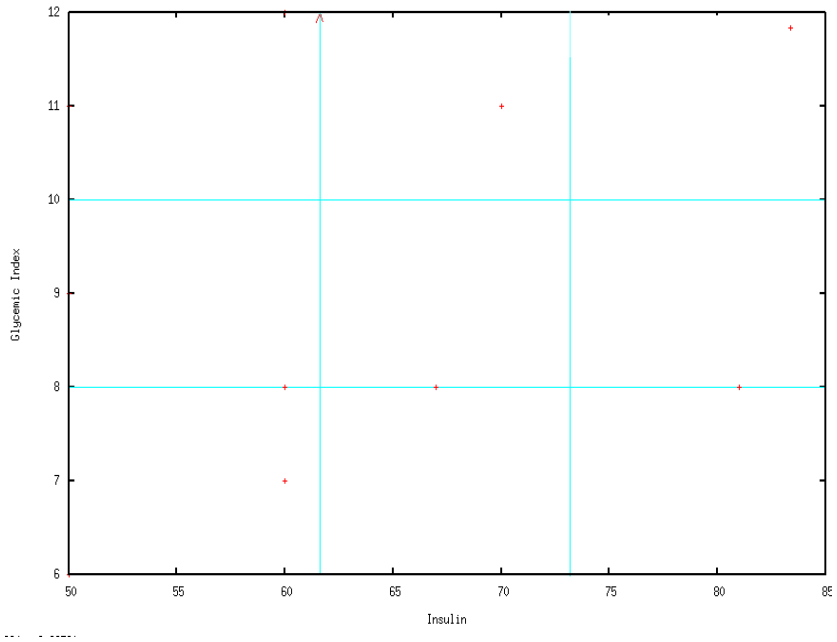

Figure 3.1: 2-dimensional data.

that in Figure 3.2 the number of data points in each bin has reduced, and the data in sparser. As the number of dimensions increases, usual distance measures become meaningless[12].

In statistical analysis, we assume linear and/or nonlinear relationships among dimensions in a dataset. A linear relationship among dimensions is explored by testing hypotheses and by measuring correlation and association of variables with the use of inferential statistical test. Typical tests include single sample t-test and single sample chi-square test. By using one of the statistical tests it can be shown that the underlying data does not conform to the linear assumption because of sparse, and high dimensional samples. Further, purely non-linear relationships amongst the dimensions also does not hold true, and we demonstrate this visually with the help of Kohonen's Self Organizing Maps [12], a popular technique used in artificial neural networks. Assume a sample table with a subset of islet cell transplant data as in Table 3.3. For training the Kohonen maps[1], we used 100 neurons and iterated 20 times. The visualization given by the Kohonen Maps is shown in Figure 3.3. Two classes XYZ and ABC from Table 3.3 are shown to be distributed unevenly and thus no clear patterns emerge using this technique.

We believe that these challenges can be approached by (a) consolidating data using pre-

[1]We used Bashir Magomedov's implementation from http://www.codeproject.com/KB/recipes/sofm.aspx

Table 3.1: Islet cell transplant dataset with Insulin measurements

PatientID	Measurement Day	Insulin (IU/ml)
XYZ	50	9
XYZ	60	8
XYZ	60	7
XYZ	50	6
ABC	50	11
ABC	60	12
ABC	70	11
ABC	67	8
ABC	81	8

Table 3.2: Islet cell transplant dataset with Glycemic Index and Insulin measurements

PatientID	Measurement Day	Insulin (IU/ml)	Glycemic Index
XYZ	50	9	53
XYZ	60	8	50
XYZ	60	7	56
XYZ	50	6	67
ABC	50	11	54
ABC	60	12	72
ABC	70	11	78
ABC	67	8	77
ABC	81	8	65

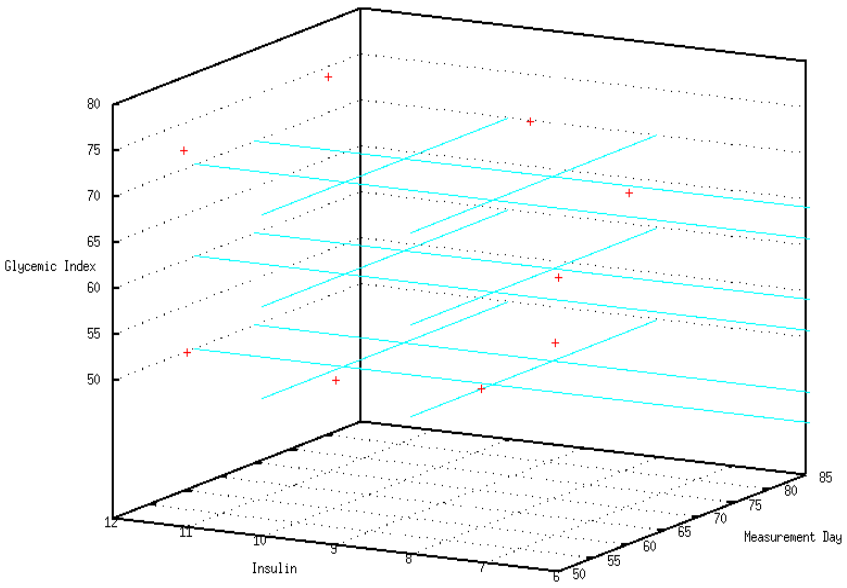

Figure 3.2: 3-dimensional data.

processing and (b) integrating statistical and data-mining techniques for knowledge discovery and (c)by providing an exploratory framework for incorporating tacit knowledge of medical experts into analysis and experimentation. Knowledge can be gained in an effective manner by integrating traditional statistical analysis tools with more sophisticated data-mining, pattern recognition and machine-learning approaches. The integration enables to explore linear and non-linear relationships during variable selection and modeling, and to exploit information present in dense and sparse samples of variables in order to build a comprehensive knowledge discovery framework for analyzing medical records. We have developed a systematic approach which consists of a flow of operations maximizing the reliability of knowledge discovery. The systematic approach involves the predictive power of data-mining techniques and statistical error analysis, the compression provided by dimensionality reduction along with the tacit knowledge of domain experts which verify the sanity of the discovered knowledge in our medical diagnosis.

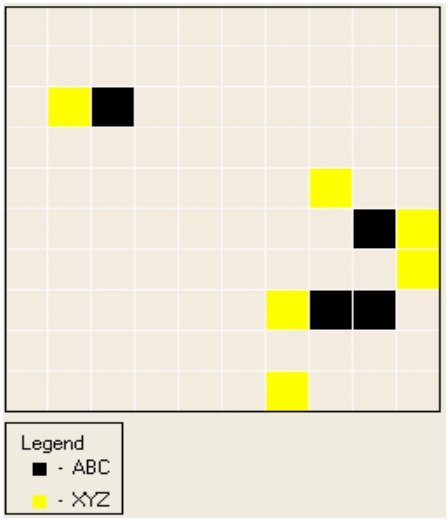

Figure 3.3: Visualization using Kohonen's Self Organizing Maps.

Table 3.3: Islet cell transplant dataset with Glycemic Index, Insulin measurements

PatientID	Measurement Day	Insulin (IU/ml)	Glycemic Index
XYZ	50	9	53
XYZ	52	10	44
XYZ	54	7	54
XYZ	56	8	60
XYZ	58	9	62
XYZ	59	6	65
XYZ	59	7	56
XYZ	60	6	67
ABC	64	11	34
ABC	66	12	42
ABC	70	11	28
ABC	72	8	77
ABC	81	8	65
ABC	50	11	54
ABC	60	12	72
ABC	70	11	78
ABC	67	8	77
ABC	81	8	65

Chapter 4

Contributions

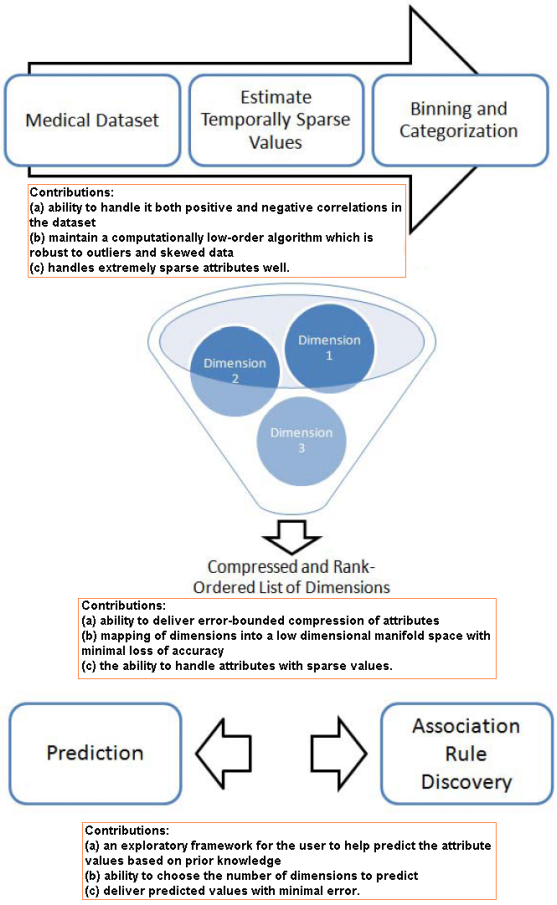

Figure 4.1: Contributions in this work.

Our main contributions (Figure 4.1) lie in (a) integrating statistical and data-mining techniques to perform several tasks enumerated in this chapter, (b) designing a methodology for dealing with sparse and high-dimensional medical records by estimation and confidence analysis and (c)creating an exploratory framework for medical diagnosis.

4.1 Data Pre-Processing

The main objective of this step is to estimate temporally sparse attribute values with high confidence given the entire measurement history of a medical patient. Usually, techniques to handle missing or temporally sparse values make three kinds of assumptions [33]: missing at random (MAR) [12], missing completely at random (MCAR) [12] and multiple imputation (MI) [34]. The MAR assumption says that the probability that a medical attribute of a patient is missing is independent of the value of that attribute itself. This means the probability of a particular medical attribute to be missing is independent of different attribute values given that all other attributes are controlled. When the probability of a missing measurement does not depend on other factors such as patient measurements during different intervals, then the phenomenon is called as MCAR. In many cases, MCAR is considered to be a subset of MAR. In [34]'s seminal work, a Monte Carlo approach [35] was used to fill up missing values with simulated values. In [34],Rubin analysed the results by standard methods and combined them to produce estimated values. Recently, techniques such as Collateral Missing Values Imputation (CMVE) [36] have become popular in that they use multiple covariance matrices for estimating missing values. Originally the CMVE algorithm was applied on gene microarray data. In our work, we extend this algorithm and adapt it to the islet cell measurements. To sum up, the main contributions of our estimation algorithm are: (a) ability to handle it both positive and negative correlations in the dataset, (b) maintain a computationally low-order algorithm which is robust to outliers and skewed data and finally (c) handles extremely sparse attributes well.

4.2 Dimensionality Reduction

Typically for large datasets and a correspondingly large number of dimensions, the "curse of dimensionality" [12] occurs as a result of which there is a rapid explosion in the amount of processing. Due to the large number of variables, the search space becomes very large leading to huge processing requirements. In this component, we compress and select a sub-set of attributes to reduce dimensionality of the final data-driven models. The islet cell transplant dataset that we have consists of a lot of dimensions, and any data mining algorithm run on this would have to endure a lot of processing and memory overhead. To circumvent this, we select a subset of

the attributes using the dimensionality reduction algorithm [37] which can then be used for further knowledge discovery. Dimensionality reduction is a problem widely studied in the machine learning literature and can be applied to both discrete and continuous variables. Dimensionality reduction could be achieved by either feature selection or feature extraction or both. Feature selection involves finding a subset of the original set of variables and feature extraction involves a mapping of the high-dimensional space into low-dimensional manifold. The algorithm designed in this work performs both feature selection as well as feature extraction. Dimensionality reduction by feature extraction consists of linear and non-linear transforms. In our work we provide an error-bounded compression with a mapping into a low dimensional manifold space with a minimal loss of accuracy and performance. In summary, the main contributions of our dimensionality reduction algorithm are: (a) ability to deliver error-bounded compression of attributes, (b) mapping of dimensions into a low dimensional manifold space with minimal loss of accuracy and (c) the ability to handle attributes with sparse values.

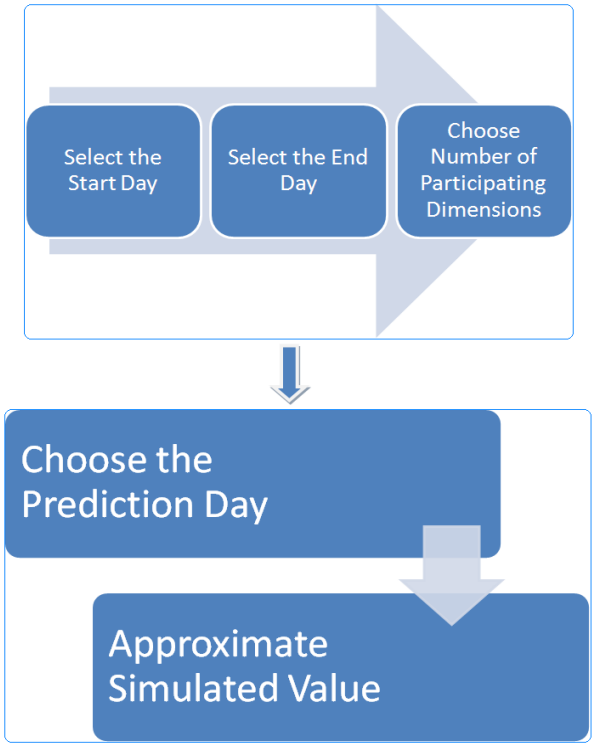

Figure 4.2: A high-level overview of the prediction algorithm.

4.3 Prediction

The knowledge discovery process is the prediction of future values of a medical patient based on prior knowledge of measurement history. Based on a prior set of patient measurements from a start date to end date in the medical history of the patient, a set of predicted values for future measurement dates are obtained by using a probabilistic bayesian-like algorithm called SDM-Prediction. Usually approaches like Maximum Likelihood (ML) and Bayesian prediction [12] are popular in statistics for estimating values. In Bayesian techniques, given an observation $X = x_{obs}$ we can calculate the probability $Pr(X = x|X = x_{obs})$ using a summary statistic from a posterior distribution. We modify and adapt the bayesian approach to handle the sparsity of the dataset that we have studied. In particular we focus on the the fact that the dataset does not have an underlying statistical model [41]. Here an alternative 'approximate' model is used in place of the original statistical model. This concept is extended by allowing us to choose the prior distribution based on the attribute ranking from the dimensionality reduction algorithm. In summary, the main contributions of our predictio algorithm are: (a) an exploratory framework for the user to help predict the attribute values based on prior knowledge, (b) ability to choose the number of dimensions to predict and (c) deliver predicted values with minimal error.

4.4 Variable Relationships

In association rule-mining (more specifically association-classification rules) we discover association rules that explore temporal relationships among measured attributes with high precision, confidence and quantifiable uncertainty[1]. Association rules satisfy minimum support and confidence values. An association rule is of the form $A \Leftrightarrow B$ are conjunctions of attribute-value pairs [12]. Our goal is to find all possible rules which satisfy minimum confidence and support threshold. Rather than mining traditional association rules which yield many unnecessary rules, we focus on developing classification-association rules following the work of [38]. Classification rule-mining is similar to association rule-mining except that the target class is known in the former. The combination technique is called as Class-association rules (CAR) first proposed by

[1]Because of the estimation of temporally sparse values, some of the attribute instances are not 'measured' but are actually 'estimated'. Association rule-discovery on 'estimated' instances may or may reflect the true nature of relationships among attributes in the dataset.

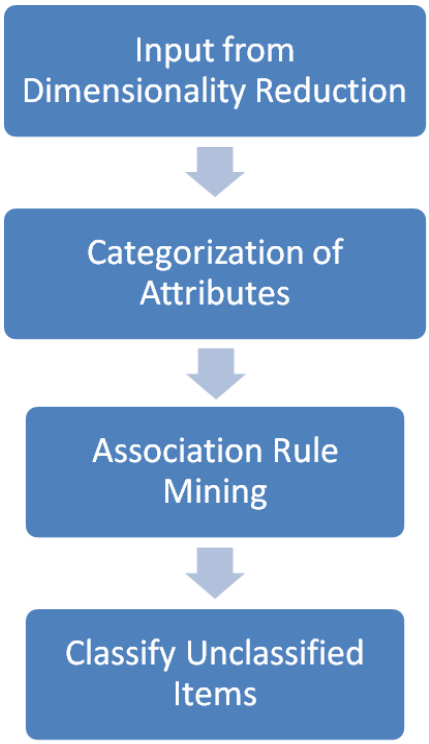

Figure 4.3: Steps involved in a generating classification-association rules.

Liu et al [38] who adapted the popular Apriori algorithm [39] in their work. This work was improved in [40] by reducing memory consumption and time complexity. Other contributions included feature subset selection and removing irrelevant itemsets. In our proposed algorithm, we take the input from the dimensionality reduction step and categorize the continuous attributes either by statistical means or by manual methods, and then apply the association rule mining and classification techniques to generate a series of "IF-THEN" rules.

Chapter 5

Theoretical Framework

In this chapter we will describe the representation of medical datasets, and the terms, definitions and a set of notations that we use to outline the algorithms in this work. After introducing the mathematical notation, the subsections follow the data flow illustrated in Figure 1.2. We precisely define the data space, the notations used while transforming data, the notations while obtaining intermediate processing results and the calculation of distance and similarity measures.

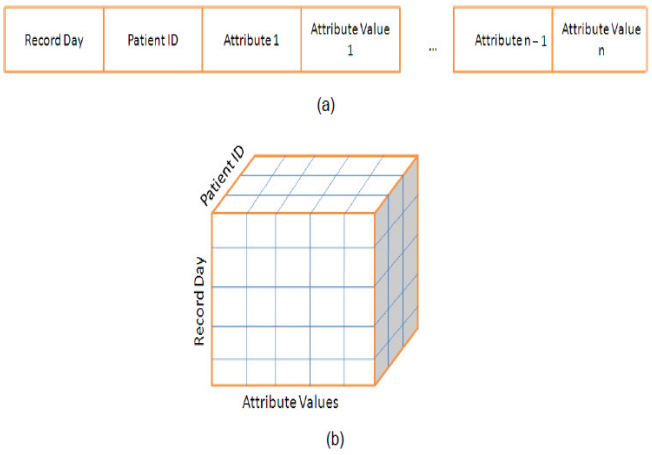

Figure 5.1: Structure of a typical record.

5.1 Record Structure

The medical dataset consists of records D_i with n variables over a set of time instances t_j. The initial data sets contain only numerical values of measurements. A typical record in the dataset is of the form as shown in Figure 5.1a. This data structure can also be represented as a multidimensional data cube as shown in Figure 5.1b. A record day could be relative or absolute. Relative records are preferred for confidentiality and for comparison purposes.

5.2 Preliminaries

D_i is considered to be an i-incomplete set of data points in an n-dimensional space. The initial set of points is i-incomplete because there are i records with temporally sparse data points. The records are in increasing order of record day, and are typically indexed by $< R_i, P_i >$ where R is the Record Day and P is the patient ID. Each R_i is a set of time instances t_{ij} where j represents an entry at a particular time of the day for record i. For processing purposes, each R_{ij} is considered independent of its set R_i. Thus D_i can be considered to be a matrix of data points. Furthermore, the i-incomplete set of Data points are transformed to k-complete set of points after data preprocessing phase. The matrix D_k is then processed using dimensionality reduction to get D_{k_i}, a set of points in the transformed space. During the data preprocessing phase, we use several statistical measures like covariance for the entire data matrix D_i to obtain a Covariance matrix C where each row has a value $Cov(D_{i,i+1})$ given by equation 5.1 below.

$$Cov(D_i, D_{i+1}) = E[(D_i - \mu_i)(D_j - \mu_j)))] \tag{5.1}$$

Now, we use the data matrix D_i to compute rank-approximations for dimensionality reduction purposes later. Here d_i and d_j are assumed to be individual row vectors and d_j^T represents the transpose of row vector d_j. Matrices $D_1 ... D_n$ are obtained in the intermediate steps while computing rank-one approximations in the dimensionality reduction process. In dimensionality reduction, we also provide a way to cluster and compress and for this we use a error measure to represent the error in the computation of a similarity distance between two intermediate row vectors. This measure is called as $Err(D)$ and is used to minimize mean squared error in

computation. More specifically, the value $Err(D_i, D_j)$ between two row vectors D_i and D_j is given by Equation 5.2.

$$Err(D_i, D_j) = ||D_i - D_j||^2 \tag{5.2}$$

In 5.2.1 we give a definition for the presence of an bounded-error transformation in the matrix space D_i. This also defines an error-bounded row vector of the data matrix, and the computation of a similarity distance between any two row vectors. Lemma 5.2.1 shows that an error-bounded row vector and transformation from Definition 5.2.1 is guaranteed to occur. To measure the usefulness of the matrices decomposed by the error-bounded function, we use the Hamming Distance metric [47]. In Lemma 5.2.2, we state that the Hamming Distance of the rank-one approximated matrix of data matrix D_i is rank-sorted in increasing order. Definition 5.2.2 defines the Hamming Distance between two row vectors d_i and d_j. We also define the Hamming Radius of a set of row vectors centered around a particular row vector in Definition 5.2.3. This Hamming Radius is used in the dimensionality reduction algorithms to decide the stopping criterion and for determining whether to partition the intermediate matrices further. The stopping criterion is adapted from [47].

Definition 5.2.1. *Let $R_{D_i} = \{R_{d_1}, R_{d_2}, ..., R_{d_n}\}$ be a set of row vectors in the data matrix space D_i. Then let $f : D_i \rightarrow \Re^k$. We call f an* error-bounded transformation *if $\forall i, j = \{1, 2..., n\}$: $Err(R_{d_i}, R_{d_j}) = \left||R_{d_i} - R_{d_j}|\right|^2$, where $\left||R_{d_i} - R_{d_j}|\right|^2$ is the mean squared error function for which there exists a bounded solution defined as the minima $M(R_{d_i}) = |R_{d_i}|^2 + \left|R_{d_j}\right|^2 - 2\,|R_{d_i}|\,\left|R_{d_j}\right|$.*

Definition 5.2.2. *The Hamming Distance between two row vectors d_i and d_j is defined as $H(d_i, d_j) = \frac{\left|d_{i_{1\times n}} \oplus d_{j_{1\times m}}\right|}{N}$, where $N = \begin{cases} n & \text{if } n > m \\ m & \text{otherwise} \end{cases}$*

Definition 5.2.3. *Given a set of rank-ordered row vectors $R_{D_i} = \{R_{d_1}, R_{d_2}, ..., R_{d_n}\}$ in the data matrix space D_i and a row vector x, the Hamming Radius of R_{D_i} centered around x is given by: $\forall i = \{1, 2, ..., n\}, H_r(R_{d_i}, x) = \ max\ H(R_{d_i}, x)$.*

Lemma 5.2.1. *Let R_v be a set of row vectors in a data matrix space D_i. Then there exists k such that the function $f : D_i \rightarrow \Re^k$ is a error-bounded transformation.*

Lemma 5.2.2. *Let $R_{D_i} = \{R_{d_1}, R_{d_2}, ..., R_{d_n}\}$ be a set of rank-ordered row vectors in the data matrix space D_i. Then let $f : D_i \rightarrow \Re^k$ be an error-bounded transformation. Let $H(f(R_{d_i}, R_{d_{i+1}}))$ be the Hamming Distance of $f(R_{d_i}, R_{d_{i+1}})$. Then the following holds true: $\forall R_{d_i} \in D_i : H(f(R_{d_i}, R_{d_{i+1}})) < H(f(R_{d_i}, R_{d_{i+k}}))$ where $k > 1$.*

Furthermore, matrix decomposition (in dimensionality reduction) is defined to be a graph partitioning algorithm as shown in Definition 5.2.4. The original graph partitioning problem was to minimize the number of different edges connecting different vertices of a k-partitioned graph. In our case, the goal is to develop a partitioning scheme which divides the data matrix into a balanced set of partitions where each leaf node consists of a set of overlapping row vectors.

Definition 5.2.4. *Given a Graph $G(V, E)$ where each V_i represents a row vector, the goal is to partition V into k subsets $V_1, V_2, ..., V_k$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$ and the number of edges of V belonging to different subsets is minimized.*

Now a regression estimate is computed in the preprocessing step and this is used to fill up the i-incomplete data matrix D_i to give a k-complete data matrix D_{k_i}. Here, given row vector d_i, the estimate over row vector d_j is given by Definition 5.2.5.

Definition 5.2.5. $d_i = \bar{d} - \frac{\theta_{d_j d_i}}{\theta_{d_i d_i}}(d_j - \bar{d_j})^2$, *where θ is an error term. In this regression model, we use parameters τ_1 and τ_1 as $\alpha = \bar{d} - \tau_2 d_j$ and $\tau_2 = \frac{\theta_{d_j d_i}}{\theta_{d_i d_i}}$. From [36], $\theta_{d_j d_i} = \frac{1}{(n-1)} \sum_{j=1}^{n} (d_j - \bar{d_j})(d_i - \bar{d_i})$.*

The dimensionality reduction algorithm uses a modified semi-discrete decomposition scheme [48] which uses a matrix transformation approach and reduces the computational complexity of the intermediate matrices generated as compared to the original scheme. Definition 5.2.6 shows the typical matrix operations present in a modified semi-discrete decomposition scheme.

Definition 5.2.6. *The decomposition of data matrix $D_i = \{d_1, d_2, ..., d_k\}$ is given by $D_{i_k} = \sum_{i=1}^{k} d_i w_i v^T$. Here w is a row vector with entries from the set $\{H(d_1, d_2), H(d_2, d_3), ..., H(d_{k-1}, d_k)\}$ from Definition 5.2.2. Also d_i is a set of positive scalars as in the original semi-discrete decom-*

position scheme and v is a column vector represented as $\begin{bmatrix} H(d_1, d_2) \\ H(d_2, d_3) \\ . \\ . \\ . \\ H(d_{k-1}, d_k) \end{bmatrix}$.

Lemma 5.2.3 states the optimal solution to the modified semi-discrete decomposition problem as a greedy approximation. This is further improved as a 'growing greedy algorithm' [49] later on in this thesis.

Lemma 5.2.3. *From Definition 5.2.6 and from [48], the decomposition of data matrix D_i, D_{i_k} be the decomposition upto the k-th term. S_t is the residual at the tth step and is denoted as $S_t = D_{t_k} - D_{t-1_k}$. The solution to finding out the contents of D_{i_k} is in finding the minima of the error-bound defined in Definition 5.2.1.*

5.3 Function-Monotonicity and Rule Measures

We develop an algorithm to mine classification association rules (or CAR) later on in the thesis. In this the main operation is to find out all rule items above a minimum support. The formal definition of CAR is given in Definition 5.3.1.

Definition 5.3.1. *From [38], the set of all items I in the transformed data matrix space D_{i_k} with a set of class labels C consists of a row vector $d_i \in D_{i_k}$ containing $x \subseteq I$ if $x \subseteq d_i$. A CAR is of the form $x \rightarrow y$, where y is a target row vector and $x \subseteq I$ and $y \in Y$. A rule D_{i_k} is of confidence c if c number of cases in D_{i_k} satisfy y.*

We now define some of the measures used in CAR and give a description of the monotone (or anti-monotone) properties of the rules generated. Definition 5.3.2 describes the monotonicity [12] of a function generated in CAR and the definitions of support and confidence while generating rules.

Definition 5.3.2. *Let z be an element of the set of rules C for data matrix d_{i_k}. Let $f : C \rightarrow \Re$ be a function associated with C such that it is a real function. Then let $<$ be an ordering relation*

on f. For any $z_i, z_j \in C$, $z_i > z_j$ implies that:

$$N = \begin{cases} f(z_i) > f(z_j) & f \text{ is monotonic on } C \\ f(z_i) < f(z_j) & \text{otherwise} \end{cases}$$

The support of a property δ belonging to C is the number of objects in C having property δ. Similarily the support of a rule $\delta \rightarrow \gamma$ in C is the number of objects in C having properties γ and δ. Confidence of a rule $\delta \rightarrow \gamma$ is given by $\frac{\text{Support } (\delta \rightarrow \gamma)}{\text{Support } (\delta)}$. Now based on this description we give a property of Confidence.

Property. Given Confidence of a rule $\delta \rightarrow \gamma$ as $\frac{\text{Support } (\delta \rightarrow \gamma)}{\text{Support } (\delta)}$. Then from [12], Support (δ) is always > 0.

The objective here is to find the items in rules that are above a minimum support count. This is represented as a set of items along with the class label that represents those set of items and is given by: $< items, c >$ where c is the class label. The items above a minimum support are *frequent* and the others are *infrequent*.

5.4 Bayes Theorem and Data Partitioning

Given $D_{i_k} = \{d_1, d_2, d_3, ..., d_k\}$ be a set of row vectors in a k-complete data matrix. From [41], if an underlying data model $g(d_i|\eta)$ represents this data matrix, then the joint probability density of D_{i_k} is given by Equation (5.3). Instead of using a fixed data model, we assume an approximate kernel mixture [41] based on a subset of the data matrix.

$$p(D_{i_k}) = \int \prod_{i=1}^{k} g(d_i|\eta) p(\eta) d\eta \tag{5.3}$$

Also by Bayes Theorem [12], the posterior distribution of η given D_{i_k} is shown by Equation (5.4).

$$p(\eta|D_{i_k}) = p(\eta|d_1, d_2, d_3 ... d_k) p(\eta) \prod_{i=1}^{k} g(d_i|\eta) \tag{5.4}$$

Given D_{i_k}, η and prior ϵ, $p(\eta|D_{i_k}, \epsilon)$ can be represented as:

$$p(\eta|D_{i_k}, \epsilon) = \frac{p(\eta|\epsilon) p(D_{i_k}|\eta, \epsilon)}{p(D_{i_k}|\epsilon)} \tag{5.5}$$

To avoid overfitting the training data in our prediction algorithm, we partition the data matrix D_{i_k} as $\{d_{i_1}, d_{i_2}, d_{i_3}...d_{i_v}, d_1^{'}, ..., d_v^{'}\}$ where d_{i_v} consists of row vectors chosen as part of the partitioned data matrix and $d_i^{'}$ consists of the row vectors which are not chosen. Chapter 10 gives a more detailed overview of the partitioning algorithm.

Chapter 6

Data Preprocessing

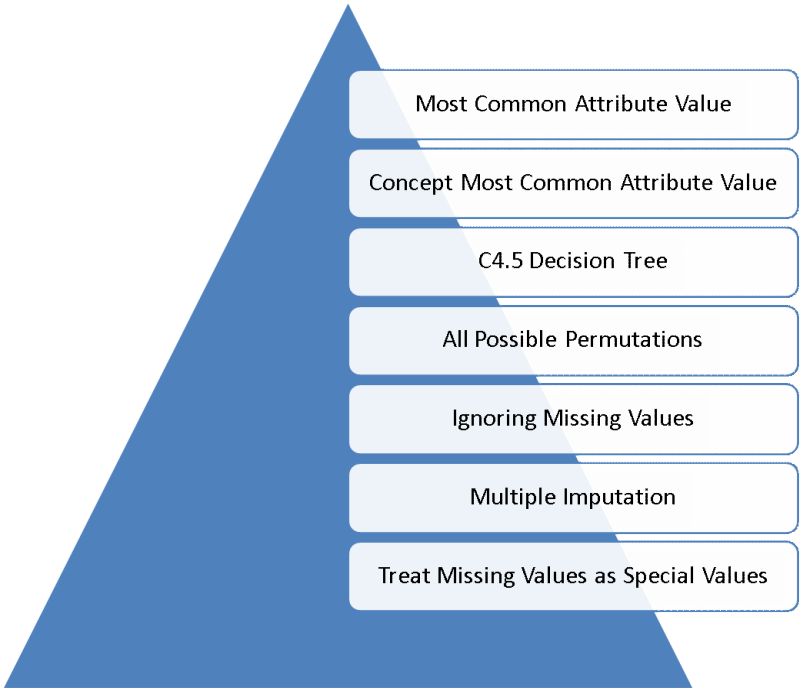

Figure 6.1: List of missing values estimation methods.

6.1 Overview of Current Techniques

To handle missing data there are techniques[12] which simply replace those missing values by zero, or by the mean/median of the dimension. Another technique is to completely ignore the missing values. To improve the performance of statistical machine learning and data mining algorithms it becomes clear that we have to reduce the prediction error while estimating missing

values. Since our dataset is temporally sparse in a high-dimensional space, it becomes an even more challenging task. Figure 6.1 shows a partial listing of the current techniques in estimating missing values. Ever since multiple imputations of missing values were published by Rubin[34], it has become increasingly clear that those methods would fail to work on datasets where there are multiple random missing values for different covariates.

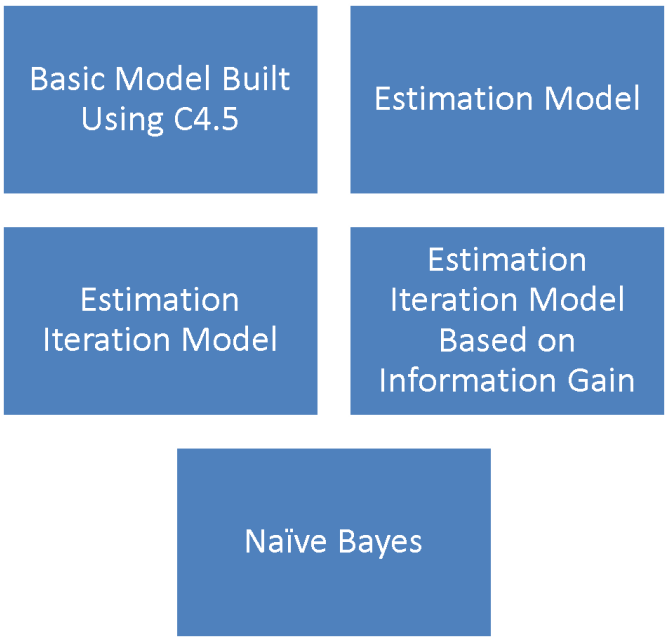

Figure 6.2: List of models built based on missing value estimation methods.

Among the techniques listed in Figure 6.1, the most common attribute method is the simplest one. In this technique, the missing value is replaced by the most commonly occurring value in that particular dimension. The next technique is to select the most commonly occurring value within a particular concept. A concept here is defined as all possible examples with the same value of the decision. The next technique is to use $C4.5$ decision classifiers [12],and uses a method of splitting the record with missing values by spreading to all possible concepts using entropy. the next technique is to use all possible values of that particular dimension to fill up the missing value. Finally we can ignore the missing values completely but if the dataset is very sparse then there would be huge variations in the results of statistical machine learning algorithms on the dataset.

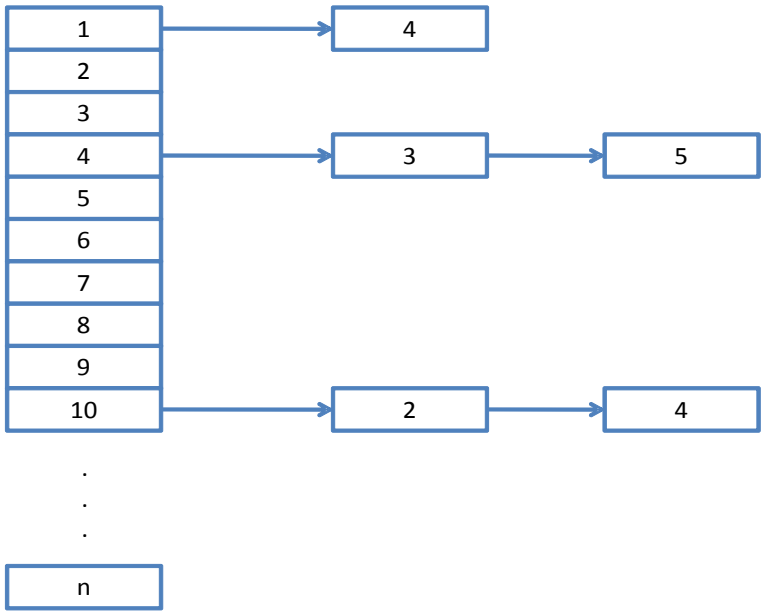

Figure 6.3: The hashed table of missing values.

Algorithm *SDM-Estimate* Hashing Algorithm

input: An i-incomplete set of data points as row vectors in matrix $D_i = \{d_1, d_2, d_3, ..., d_n\}$ in an n dimensional space S. Also given an empty index structure H of size i. Given hashing function HASH and r as hashed value. **output**: A complete index structure H

begin

1. $r =$ `HASH`(i, n)
2. `Repeat the following while ( (`$H[r]$ `is occupied) and (` $H[r].i \neq i$`))`
3. $r = (r + 1)$ `modulo` n
4. `if (`$H[r]$ `is occupied)`
5. `flag =1`
6. `End if`
7. `End While`
8. `if (flag =1)`
9. $H[r] = j$
10. `End if`

end

Figure 6.4: *SDM-Estimate* hashing algorithm

Apart from the techniques mentioned above, there are a few other algorithms which attempt to predict missing values using machine learning. The first is $k - NN$ algorithm[12] which tries

to estimate missing values by filling them with values similar to the missing value. The similarity measure is given by the Euclidean distance metric. However it cannot handle sparse data well as the presence of outliers may not be easily detected by the algorithm. The next technique is by using PCA-based estimation [53]. The main advantage of this technique is that the computation complexity of the algorithm is $O(mn)$ but speed is not considered as a very important criteria in missing value estimation.

Figure 6.2 shows a list of machine learning models which are applied on the dataset based on the techniques for handling missing values [54]. In the basic model, the missing values are ignored completely. In the second model, bayesian classifiers are used to build a model and to fill up all missing values. In the third technique, bayesian techniques are used for estimation but values obtained in prior iterations are not used for estimation purposes in the current iteration. In the bayesian estimation technique with information gain, only the first iteration is used as training subset and finally the naive bayes classifier is used as the fifth model. The authors found out that the bayesian techniques based on information gain would achieve the best performance for large datasets without any significant overhead and costs.

6.2 SDM-Estimate Algorithm

This algorithm is inspired by [36] where the authors developed a multiple covariance matrix based imputation algorithm for estimating missing values in gene microarray data. SDM-Estimate predominantly makes uses of the covariance measure as it handles both positive and negative correlation.

A list of missing values in each vector is kept in a hash table as shown in Figure 6.3. This hash table consists of indices ordered by rows, and buckets ordered by columns with each bucket containing a value denoting the column number of the missing value in that vector. The hashing algorithm is shown in Figure 6.4. We use a chaining hash mechanism with linear probing and single-slot stepping. In our hashing algorithm, we search for an empty slot by iterating through the index using the HASH function and continue till a slot is found. If an empty slot is not found we enlarge the index H by $n + K$ where K is each unsuccessful iteration.

In case of a collision at position r we add the current missing value to the end of the (key,value)

pair at r as a linked list. Let the row vector of the missing value be H_{ij} (from the hash table). To locate a missing value from this index, Figure 6.5 shows the LOCATE algorithm. For an n-dimensional data matrix $D_i = \{d_1, d_2, ...d_n\}$, the covariance of each row vector d_i with respect to the d_m-the row vector represented by H_{ij}, is given by Definition 5.1. Accordingly it is computed for all row vectors in data matrix D_i. The covariance is stored in C and is ranked in ascending order.

Algorithm *LOCATE* algorithm

input: An i-incomplete set of data points as row vectors in matrix $D_i = \{d_1, d_2, d_3, ..., d_n\}$ in an n dimensional space S. Also given a complete index structure H of size i. Given hashing function HASH and r as hashed value
output: Position x of next missing value in H
begin
1. $r =$ `HASH`(i, n)
2. `Repeat the following while ( (`$H[r]$ `is occupied) and (` $H[r].i \neq i$`))`
3. $r = (r + 1)$ `modulo` n
4. `if (`$H[r]$ `is occupied)`
5. `flag =1`
6. `End if`
7. `End While`
8. `if (flag =1)`
9. $x = r$
10. `Return` x
11. `End if`

end

Figure 6.5: *LOCATE* algorithm

The row vector D_k corresponding to the topmost index of the covariance matrix C (or C_0) is used to compute μ_1, μ_2, μ_3 and μ_4 in Equations (6.1), (6.2), (6.3) and (6.4) respectively.

$$\mu_1 = \tau_1 + \tau_2 \cdot D_k + \tau_3 \cdot D_{k+1} \tag{6.1}$$

Here a least-squares regression method is used to estimate μ_1 and the values of τ_1 and τ_2 are specified in Definition 5.2.5. From [36], τ_3 is used to reduce the error in the computation of μ_1 and is considered as an "error term".

$$\mu_2 = \frac{\sum_{i=1}^{k} \mu + \eta - \sum_{i=1}^{k} \kappa^2}{\eta} \tag{6.2}$$

$$\mu_3 = \frac{\sum_{i=1}^{k} (\mu^T \times \texttt{TEMP})}{k} \tag{6.3}$$

Algorithm *SDM-Estimate*

input: An i-incomplete set of data points in matrix $D_i = \{d_1, d_2, d_3, ..., d_n\}$ in an n dimensional space S. Also given hash index H.
output: K-complete set of data points D_K in S
begin
1. Set $K = 0$
2. Repeat while $k < n$ // k **is the index of the current row vector**
3. Repeat while LOCATE($H[i] \neq -1$)
4. set TEMP $= d_{ij}$ // **Here TEMP shows the row vector (with index i) for a missing value j**
5. End While
6. Calculate covariance (using Definition 5.1 of d_k and TEMP as $Cov(d_k, \text{TEMP})$
7. l=0 // l is the index of the covariance array
8. Repeat while ($l < n$)
9. Store $C_l = Cov(d_k, \text{TEMP})$
10. Rank D_i based on C_l
11. $l = l + 1$
12. End While
13. Select D_k corresponding to C_0
14. Use D_k to compute μ_1
15. Calculate μ_2, μ_3 and μ_4
16. Use $\gamma = T_1 \cdot \mu_1 + T_2 \cdot \mu_2 + T_3 \cdot \mu_3 + T_4 \cdot \mu_4$ as calculated missing value for d_{ij}
17. Update d_{ij}
18. $K = K + 1$
19. End while
end

Figure 6.6: *SDM-Estimate* algorithm for estimating temporally sparse values

$$\mu_4 = \frac{\sum_{i=1}^{k} (\mu^T \cdot \kappa)}{\mu_2} \tag{6.4}$$

$$\tau_3, \mu, \eta = min(\kappa) \tag{6.5}$$

Here η and μ are the normal residual and actual residual respectively. The objective function

in Equation (6.5) minimizes the prediction error κ using non negative least squares estimation technique[55]. According to [55], this problem can be solved by iterations which always converge and terminate. The iterations may take a long time to converge but the final solution is "fairly good". Algorithms for non negative least squares estimation can be divided into two types of approaches: *active set* and *iterative approaches*.

We follow an iterative approach [56] where the current solution is updated with the help of projected gradient methods where the update takes place towards the steepest descent. The projected gradient approach used to minimize the objective function in Equation (6.6) for the non negative least squares estimation technique. Finally the missing value estimate is given by γ in Equation (6.7).

$$min|\,|D_i - D_k||_2 \; s.t. each \; d_{ij} \in D_i \geq 0 \; and \; \kappa = D_i - D_k \tag{6.6}$$

$$\gamma = \tau_1 \cdot \mu_1 + \tau_2 \cdot \mu_2 + \tau_3 \cdot \mu_3 + \tau_4 \cdot \mu_4 \tag{6.7}$$

Note that in our data preprocessing step we only consider the absence of attribute values, and not the absence of attributes themselves. An additional step of standardizing the dataset by domain experts is an important step and is currently part of ongoing research work. Another important step of detecting outliers in data is also necessary to remove attribute values which may skew the performance of prediction and rule-mining algorithms. This is also part of ongoing research, and is quite a challenging problem because of the sparse nature of data.

6.3 Summary

This chapter describes the SDM-Estimate algorithm, which estimates temporally sparse values using a combination of hashing and multiple correlation estimation techniques. An iterative approach is followed to minimize errors in estimation.

Chapter 7

Dimensionality Reduction

7.1 Overview

Dimensionality reduction techniques can be divided into feature extraction and feature transformation techniques. Feature transformation techniques transform the original data space into fewer dimensions by combinations of the original attributes. The main disadvantage of these techniques is that the new features are sometimes difficult to interpret in the transformed domain. Feature extraction techniques select only the most relevant dimensions from the dataset which are representative of the entire dataset. Feature extraction techniques are ineffective when the data points are spread out in multiple dimensions.

Dimensionality Reduction Techniques can be categorized into linear and non-linear techniques [37]. Among the most widely used linear technique is Principal Component Analysis (PCA) [12]. PCA is also known as Singular Value Decomposition (SVD), the Karhunen-Loueve transform, the Hotelling transform, and the empirical orthogonal function (EOF) method [58]. PCA finds an orthogonal transformation of the original dataset that combines variables with the largest variance in the dataset. Factor Analysis [57] is another technique that makes an assumption that measured variables in a dataset depend on unknown and often unmeasurable factors. This helps in reducing variables to a low-dimensional form using a factor model. Two methods of deriving certain model parameters in the Factor Analysis technique is by Principal Factor Analysis and by Maximum Likelihood Factor Analysis.

Another major category of techniques is by using Projection Pursuit[59]. This technique can incorporate higher than second-order information. Projection Pursuit looks for the most interesting directions in a projection index (A projection index maps from higher order dimensions to a lower order). Yet another technique is Independent Component Analysis (ICA)[61] which looks

for projections that are as statistically independent as possible. Among the non-linear techniques, a useful technique is the non-linear independent component analysis [60]. Another technique is finding Non-linear Principal Curves that run through a multivariate dataset. Multidimensional scaling [37] is another technique which finds a matrix representation of the original dataset in a lower dimension that preserves the proximities between the items. Kohonen self-organizing maps [12] are yet another technique that transform a higher-dimensional dataset into a lower dimensional fixed lattice. Density Networks, Neural Networks and Vector Quantization Techniques [37] are also representative non-linear techniques. The last major category of algorithms are the ones based on genetic and evolutionary computation.

7.2 Problem Statement

The dimensionality reduction problem can be formulated as follows:

Problem 1. *Given h row vectors d_1, d_2,..., d_h from data matrix D_i in an n-dimensional space, the goal is to find k row vectors r_1, r_2,..., r_k such that the following holds true:*

$\forall 1 \leq i \leq h, \exists j$ *s.t.* $|Err(d_i - r_j)|^2 \leq \chi$, *where χ is a tight error bound, represented as a Hamming distance metric(from Definition 5.2.2).*

7.3 SDM-Reduction Algorithm

We have designed an indexing mechanism to dimensionality reduction inspired by [47] with the following important characteristics:

- It discovers representative patterns in the data using a partitioning-based algorithm.
- It uses a multi-resolution indexing mechanism to cluster data.
- It preserves the distance measures in the transformed space.
- It provides a rank-one approximation of the original matrix using the multi-layered indexing structure.

Following the work of [47], the DR problem can be solved by finding a discrete rank-one approximation of the input matrix formed from all medical records (one time instance per row and

one variable per column). We represent each row vector in a multi-dimensional ring structure, called as a "peer". This ring structure, called as *MLR-Index* [62] is organized in a concentric manner and is space-efficient. The rank-one approximation is obtained by computing the Hamming Radius (Definition 5.2.3) of each row vector with its neighboring row vectors (in the ring). The minimal error computations (Definition 5.2) are then used to order the row vectors and determine the rank-one approximation matrix.

From our previous work [62], we create the MLR-Index using a partitioning algorithm, shown in Figure 7.3. After we partition the data using the partitioning algorithm, we develop a search algorithm [12] to find out the row vectors most similar to the current row vector. To develop the concentric ring based index structure we use a representative row vector called as a "search node", which keeps track of $O(logN)$ row vectors. The concentric rings are of exponentially increasing radii. In Definition 7.3.1(from [62]), we describe the structure of the multi-dimensional index.

Definition 7.3.1. *Define V to be a high-dimensional vector space $\Re^d$ where $d > 20$ and D to be a finite set of data points where $D \subset V$. Define $dist(p, q)$ to be the Euclidean distance between two points p and q in V. Define $B_p(r)$ to be a multi-dimensional set of points with radius r centered at p in V.*

Now, given a row vector as a query point q in this transformed space, we formulate the search problem as shown in statement 2.

Problem 2. *Given a constant k and a row vector q, find k-nearest row vectors of q in D.*

In V, we choose N row vectors that help in performing an efficient search algorithm. Definition 7.3.2(from [62]) describes a search node in MLR-Index.

Definition 7.3.2. *Let C be a cluster of data points in D and p be its center. We say p is a search node. Denote $C(p)$ to be C and S to be a set of all search nodes. Define radius $(C(p))$ as the maximum distance between the center p of the cluster $C(p)$ and the data points in $C(p)$. That is, $radius(C(p)) = max(dist(p, q)|q \in C(p))$. Define $size(C(p))$ as the number of data points in a cluster $C(p)$.*

Figure 7.1(from [62]) shows the multi-layered index structure.

Our first goal is to find the closest row vectors (also called as "nearest neighbors [12]) of a given row vector. To perform this task we perform a search on each search node (S) first,

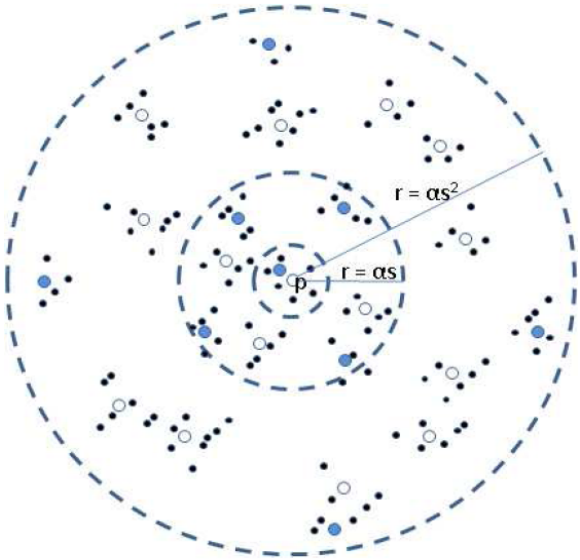

Figure 7.1: The multi-layered index structure.

and then refine the search to other row vectors. A search node retains a ring structure for this procedure, and sometimes we use additional data structure, a list of the m nearest row vectors, for a faster search. A list of m-nearest row vectors of p is given in Definition 7.3.3.

Definition 7.3.3. *For a search node p, we define $m - NS(p)$ as a list of m nearest row vectors of p.*

To find out the nearest row vectors to each row vector, we develop a technique which progressively reduces the search steps in each successive computation (More specifically, it is of $O(logN)$ steps). In the initial stages each search node keeps track of a small, fixed number of other search nodes in V. To further refine the search, a list of other row vectors is developed into concentric, non-overlapping rings. This ring structure favors nearby neighbors by providing information on search nodes in the immediate vicinity. The formal representation of a MLR-ring is shown in Definition 7.3.4(from [62]).

Definition 7.3.4. *For a search node p, the i-th ring has inner radius $r_i = \alpha s^{i-1}$ and outer radius $R_i = \alpha s^i$ for $0 < i < i*$ where $i*$ is a user defined parameter. For the innermost ring with $i = 0$, we define $r_0 = 0$ and $R_0 = \alpha$. All rings with $i \geq i$ are collapsed into a single outermost*

ring with $r_i = \alpha s^{i*}$ *and* $R_{i*} = \infty$.

Now the MLR-Index is formally described in Definition 7.3.5(from [62]).

Definition 7.3.5. *We define MLR-Index to be a set of all search nodes together with their ring structures and radius values. For a search node p, we denote MLR-Index(p) to be p's ring structure together with* $radius(C(p))$.

To find all nearest row vectors to a particular row vector q we begin with finding the nearest search node p of q (known as the nearest search algorithm in Figure 7.2). To bring the search closer to the nearest search nodes of q, w first randomly choose a search node p and measure the distance between p and q. To refine the search further, we compute the minimum distance between q and the search nodes in MLR-Index of p. This distance function is given in Definition 5.2.2. If we find a closer search node, then find the minimum distance between its index structure and q. This procedure is repeated until we cannot find any closer search nodes. By repeating the procedure described above m times, we can find m-th nearest search node of q. From [62], the nearest search algorithm is shown here for finding the nearest search node of q.

Algorithm *Nearest search algorithm*
input: A query row vector q
output: Nearest search node (row vector) p of q
begin
1. `Randomly choose any search node` p
2. $d = dist(p, q), d \leftarrow \infty$ `(From Definition 5.2.2)`
3. $\hat{d} = d$
$d =$ `the minimum distance between the search nodes`4.
5. `if` $(d < \hat{d})$
6. $p = MinP_d$
7. `Output` p
end end

Figure 7.2: Nearest search algorithm

Figure 7.3 shows the algorithm for partitioning. First, it finds the nearest search node p. If the innermost ring of p contains more than or equal to k row vectors, then find the kth nearest row vector of q within the innermost ring of p. If the innermost ring of p contains less than k row vectors then execute the nearest search algorithm until the next nearest search nodes of q

contains at least k row vectors. Then find the kth nearest row vector of q within A.

Algorithm Partitioning algorithm

input: A row vector q and constant k
output: Closest row vectors of q
begin
1. $p =$ `Nearest Search`(q)
2. `if` $|C(p)| \geq k$
3. $d = dist(q,$ `k`$)$`th nearest row vector of` q `in` $C(p)$`)(From Definition 5.2.2)`
4. `else`
5. $A = C(p)$
6. $S = S \cup \{p\}$
7. `While` $|A| < k$
8. $\hat{p} =$ `Nearest Search`(q, S)
9. $S = S \cup \{p\}$
10. $A = C(\hat{p})$
11. $d =$ `dist(`$q,$ `k`$)$`th nearest row vector of` q `in` A`)`
12. `Output closest row vectors of` q `within` $\hat{C}$

end

Figure 7.3: Partitioning algorithm

From [62], Theorem 7.3.1 proves that our partitioning and nearest search node algorithm always find the k closest row vectors to a give row vector in $O(logN)$ steps.

Theorem 7.3.1. *Let q be a row vector and p_i be the ith nearest search node of q. Let R be the maximum radius of all clusters. Let j be the smallest number such that $\left|\bigcup_{i \leq j} C(p_i)\right|$ contains at least k row vectors. Let o be the kth nearest row vector of q in $\left|\bigcup_{i \leq j} C(p_i)\right|$ and d be the distance between o and q. Suppose that m is the smallest number such that $dist(q, p_m)$ becomes bigger than $R + d$. Then, there exists no i which is bigger than m such that $C(p_i)$ contains any of the k closest row vectors of q.*

Proof. Suppose $i > m$ and that there exists a row vector $\hat{o}$ in $C(p_i)$ which is one of closest row vectors of q. Then, $dist(q, \hat{o}) \geq dist(q, p_m) - R > d$. But $B_q(d)$ already contains at least k row vectors. This results in a contradiction. Hence, proved. □

Figure 7.4 gives an algorithm for refining the search algorithm of Figure 7.2 using $m - NS$ search nodes (Definition 7.3.3). At first, we execute the nearest search algorithm to find the nearest search node p from query point q. If the innermost ring of p contains more than or equal

Algorithm Partitioning and search algorithm using $m - NS$ data structure

input: q and k
output: Closest row vectors of q
begin
1. $p =$ `Nearest search`(q)
2. `if` $|C(p)| \geq k$
3. $d =$ `dist(`q, k`th nearest row vector of` q `in` $C(p)$`)`
4. $\hat{C} =$ `Closest row vectors of` q `within` $C(p)$
5. `for` $\hat{p} \in m - NS(p)$
6. $C = \hat{C} \cup C(\hat{p})$
7. `else if` $|$`clusters of` $m - NS(p)| < k$

`apply Nearest search algorithm repeatedly until we get` $\hat{m}$ `such`8.

9. `else`
10. $A = C(p)$
11. $S = S \cup \{p\}$
12. `While` $|A| < k$
13. $d =$ `dist(`q, k`th nearest row vectors of` q `in` A`)`
14. `Output Closest row vectors of` q `within` $\hat{C}$

end

Figure 7.4: Partitioning and search algorithm using $m - NS$ data structure

to k row vectors, then we find the kth nearest row vector of q within the innermost ring of p. If the innermost ring of p contains less than k row vectors but the union of clusters A of $\hat{m} - NS(p)$ contains more than or equal to k row vectors, then find the kth nearest row vector of q within A.

7.4 Summary

In this chapter, we have described SDM-Reduction, an indexing mechanism to dimensionality reduction which discovers representative patterns in the data, forms clusters and provides an approximation of the original data using a multi-layered indexing structure.

Chapter 8

Rule-Mining and Prediction

To illustrate knowledge discovery, we selected two data mining techniques, such as the rule association based discovery of variable relationships, and Bayesian prediction.

8.1 Generating Classification-Association Rules

8.1.1 Overview

From [12], generally an association rule is of the form $A \Rightarrow B$, for $A, B \subseteq C$, where C is a set of all items, and A and B are itemsets. Now, there could be a number of records considered to have A as subsets. Let these records be denoted by $freq(A)$, and let T be the total number of records, then (from Definition 5.3.1)

$$Support(A) = \frac{freq(A)}{T}. \tag{8.1}$$

Confidence and Support of a rule(from Definitions 5.3.1 and 5.3.2) are given by:

$$Confidence(Rule) = \frac{freq(A \cup B)}{freq(A)} \tag{8.2}$$

$$Support(Rule) = \frac{freq(A \cup B)}{T}. \tag{8.3}$$

The goal of generating CAR rules [38] is find out the set of rules satisfying a minimum support and to build a classifier for the rules. The description of CAR rules is given in Definitions 5.3.1, 5.3.2. To generate the rules, a concept of "frequent" and "infrequent" itemsets is used with the rule items satisfying a minimum support known as frequent and the others as infrequent.

The implementation is inspired by [63] in that we use an Apriori rule-mining algorithm and generate classes of unseen samples based on the target items collected from the rules. We improve upon this criterion by including an threshold support criteria called as "iceberg support". We select only those rules for classification whose instances satisfy a minimum number of examples as specified by the iceberg support. Our contributions are listed below.

- Provide an iceberg support criteria to prune rules not satisfying a minimum number of examples.
- Improve the APRIORI-C algorithm [63] for generating rules by maintaining lower space and time complexity.
- Improve the accuracy of the APRIORI-C algorithm by generating a balanced number of classes across the dataset.

8.1.2 Algorithm

We first preprocess the data matrix $D_{k,L}$ using binning and user-guided categorization in order to make it suitable for association rule mining. We use binning by means, by frequency and by median values [12]. User-guided categorization consists of manually labeling the dataset into different Range,Label values. In our rule mining process we first generate the candidate set of rules using the Apriori algorithm. We obtain the support of each k-frequent candidate itemset for inclusion in the final rule set. We then generate $k+i$ itemsets which are k-frequent, and in this way the complete data matrix D_{ki} is covered. In the next step we use an iceberg support criteria called as I_S to prune all infrequent itemsets from D_{ki}. The description of the iceberg support criteria is given in Definition 8.1.1.

Definition 8.1.1. *Let r be a ruleset consisting of T target items. Then $\forall T \in N$, where N is the total number of dimensions, and iceberg property I_S (which is anti-monotonic), r can be partitioned(or pruned) as follows: $P_x(r) = \{r_x | I_S / r_x \in r\}$.*

After generating the association rules, we classify the unclassified examples in the dataset using the target rules in the algorithm. For each rule in the list of discovered rules, consider a candidate l-itemset. Incrementally traverse the rule items and obtain the support of each

Algorithm *SDM-Rules*
input: A (k, L)-complete set of data points, dimensions $D_{k,L}$ in a transformed space S_i;
given iceberg support I_S
output: A set of association rules D_R in S_i
begin
1. For each set of row vectors $d_i \in D_{k,L}$
2. For each j
3. Categorize d_{ij} by using Binning or User-guided Categorization
4. End for
5. Get support S of each i-itemset
6. Compare S with minimum support and obtain set of frequent i-itemsets
7. Generate candidate k-itemsets using Apriori property in D_{ki}
8. Obtain support of each candidate k-itemset for inclusion in the final set
9. Generate $k + i$ itemsets which are k-frequent
10. End for
11. For each target item T in D_{ki}
12. l = 1
13. C_l = set of all l-itemsets
14. Check the iceberg support of all itemsets in C_l with I_S
15. Prune items from C_l which are not supported by I_S
16. For Each itemset in C_l do
17. Generate candidate k-itemsets using Apriori property in C_l
18. Obtain support of each candidate k-itemset for inclusion in the final set
19. Put them into C_{l+1}
20. if $|T| = 1$
21. Put C_l in D_R
22. End if
23. $l = l + 1$
24. End for
25. End for
end

Figure 8.1: *SDM-Rules* algorithm for discovering classification association rules.

candidate itemset. If the support of target item T in each candidate l-itemset is 1 then we classify the examples in dataset D_{ki} using T. If classification cannot be achieved by T, we incrementally keep adding candidate itemsets to the current rule until a minimum number of examples can be classified.

Memory consumption is taken care of in SDM-Rules by removing all infrequent itemsets using the iceberg support criteria I_S. Time taken to compute the supported itemsets satisfying the minimum number of examples required to belong to a particular class is also reduced as supported itemsets of k and $k+1$ need not be scanned again after pruning using the iceberg support criteria.

The original APRIORI-C algorithm itself provides low memory overhead, and by ignoring these two itemsets, the memory space is reduced further. If a target item does not satisfy a minimum number of cases then the algorithm keeps on incrementally adding more candidate rules until the support count can be satisfied.

8.2 Bayesian Prediction

We adapt a Naive Bayes classifier [12] that has been modified to process numeric data [64]. Let x be a row vector we want to classify, and c_k be a possible class (label). What we want to know is the probability that the vector x belongs to the class c_k. We first transform the probability $P(c_k|x)$ using Bayes' rule: $P(c_k|x) = P(c_k) \times \frac{P(x|c_k)}{P(x)}$

Class probability $P(c_k)$ can be estimated from training data. However, direct estimation of $P(c_k|x)$ is impossible because of the sparseness of training data. By assuming the conditional independence of the variables forming medical records elements and used for constructing a vector x, $P(x|c_k)$ is decomposed as: $P(x|c_k) = \prod_{j=1}^{d} P(x_j|c_k)$, where x_j is the jth element of vector x. Then the previous equation becomes: $P(c_k|x) = P(c_k) \times \frac{\prod_{j=1}^{d} P(x_j|c_k)}{P(x)}$.

With this equation, we can calculate $P(c_k|x)$ and classify x into the class with the highest $P(c_k|x)$. Note that the naive Bayes classifier assumes the conditional independence of features which is not always the case with different features of the islet cell dataset. Inspite of this, the naive Bayes classifier exhibits good performance in general.

8.3 Summary

In this chapter we have provided an overview of predicting variable relationships using rule-mining and prediction algorithms. Our rule-mining algorithm discovers classification-association rules while the prediction algorithm uses a version of the bayesian classifier for prediction.

Chapter 9

Experimental Results

In this section, we describe our experimental results over real and synthetic data.

9.1 Data

The real dataset consist of islet cell transplant data. The synthetic data is generated by sinusoidal functions, and it validates the effectiveness of our algorithms. We define the sinusoidal functions used in our experiments as follows: $x(t) = cos(\omega_0 t)$ where $w_0 = n\pi$. The spectrum of this sinusoidal signal is given by $X(\omega) = \delta(\omega - \omega_0) + \delta(\omega + \omega_0)$. To sample the signal $x(t)$ is multiplied by $p(t)$, and its frequency equivalent becomes $X_p(\omega) = X(\omega) * P(\omega)$. The base sampling frequency is given by $\omega_s = \frac{1}{T_s}$. We update this sampling frequency as $\omega_s = n\frac{1}{T_s}$ for experimental purposes.

9.2 Evaluation Metrics and Results

We define a few evaluation metrics to demonstrate dependencies of data quality on the confidence in knowledge discovery results. Two sets of metrics are defined, one for prediction and one for association rule discovery. The first metric is the mean error on predicted values from these datasets. It is shown in Equation (9.1)below.

$$\sum_{i=1}^{n} \frac{Actual_{Value} - Predicted_{Value}}{n} \tag{9.1}$$

The main purpose of the mean error on predicted values is to determine the effectiveness of our prediction algorithm. The next set of metrics is for evaluation of association rule discovery. The first metric here is Precision. Precision measures the degree to which instances actually satisfy the rule (in the presence of uncertain values). Some of the instances have a degree of

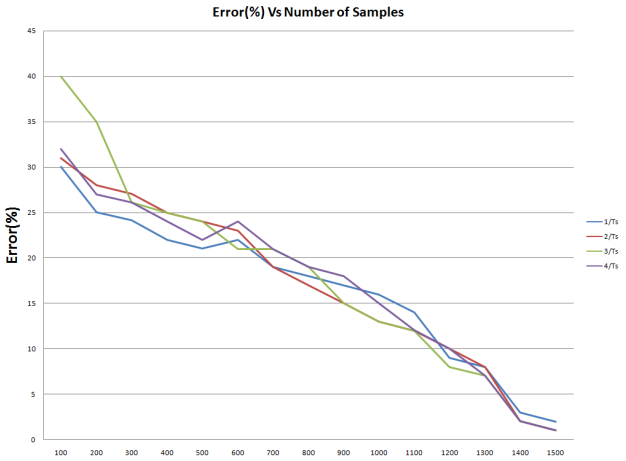

Figure 9.1: Mean Error(%) Vs Number of Samples for Baseline Synthetic Case.

Table 9.1: Results of association rule discovery

Sample Size	Support	Precision	Confidence
20	50	56	55
30	60	57	51
40	60	61	65
50	50	65	78
60	50	65	79
70	60	78	81
80	70	74	84
90	67	81	86
100	81	85	88

uncertainty attached to them because of the estimation of some values for certain dimensions. Precision is defined as shown below.

$$Precision = \frac{\sum_{i=1}^{n} \text{No. of uncertain instances}}{\text{Total no. of instances satisfying the rule}} \tag{9.2}$$

For measuring prediction error in baseline case for each instance, we measured the difference between the estimated value (using our SDM-Estimate algorithm) and the predicted value (with the entire measurement history of a patient considered as prior knowledge). It is given as: $Error_est\% = \frac{EstimatedValue - PredictedValue}{EstimatedValue} \times 100$. In cases when the 'measured' value is present

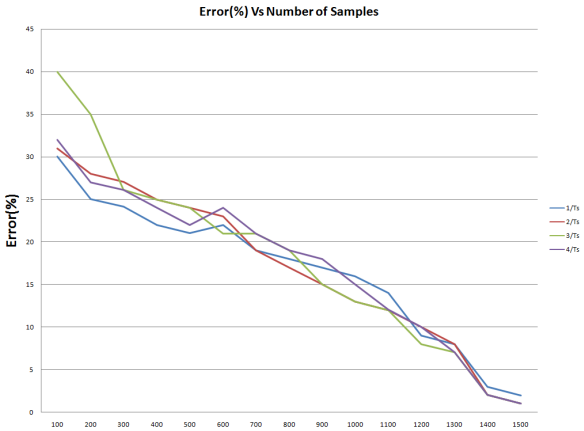

Figure 9.2: Mean Error(%) Vs Number of Samples for Random subsampling.

the prediction error becomes: $Error_meas\% = \frac{MeasuredValue - PredictedValue}{MeasuredValue} \times 100$.

Uncertainty of an instance satisfying a discovered rule is assigned as 'High' if the number of uncertain dimensions in an instance is $\geq \frac{2}{3}N$ where N is the total number of dimensions in that instance. It is assigned as 'Low' otherwise. The other two measures are support and confidence of a rule.

Three types of experiments are performed on synthetic datasets. The first is the baseline case, when the experiments are performed on the original synthetic dataset. The second set of experiments is by the introduction of Gaussian noise in the dataset. The third experiment is by removing dimension values from the dataset to simulate the absence of values. The process of removing is done by a random sub-sampling process. For each case the mean prediction error is reported as a function of sampling rate. For the real dataset, only the baseline case is considered.

For association rule discovery, we report the results of cross-validation on the real dataset with precision, support and confidence of the rules. Figure 9.1, 9.2 and 9.3 show the results of prediction on synthetic data for each of the experimental design cases. We can see in Figure 9.1 that as the sampling rate increases, the mean prediction error decreases (more samples implies smaller error). This is the result of the Bayesian prediction algorithm where the probability of a particular value is the product of both likelihood and prior probability, and the value increases with the number of available samples. In Figure 9.2, the mean prediction error for various σ

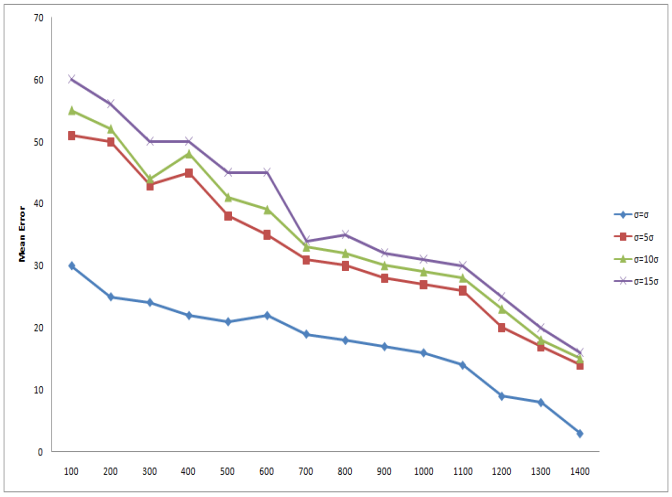

Figure 9.3: Mean Error(%) Vs Number of Samples for Various σ of Gaussian noise.

values of Gaussian noise are given. For a fixed sampling rate, the values of mean prediction error increase with increasing σ (more noise implies larger error). Finally, in Figure 9.3, we can see a curve which shows the results of periodic sampling and random sampling for a given sampling rate.

Figure 9.1 shows the results of association rule discovery on the real dataset. For a given support threshold and cross-validation fold, the precision and confidence are tabulated. The table shows that precision and confidence are consistent are less for a lower support threshold but gradually stabilize to consistent values for larger support thresholds.

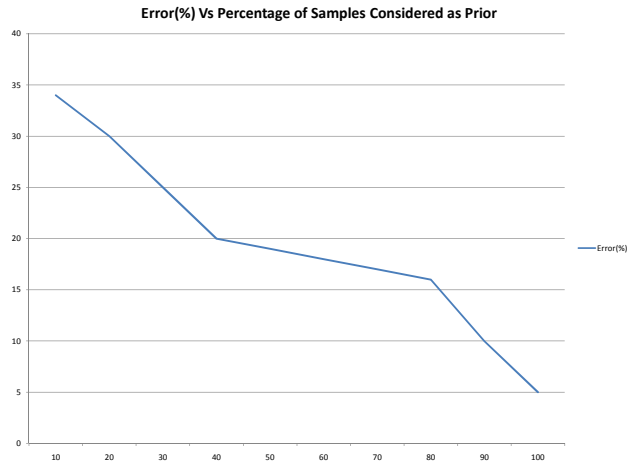

Figure 9.4: Mean Error(%) Vs Percentage of Samples considered as Prior Knowledge for Real dataset.

Chapter 10

Conclusions and Future Work

In this work we presented SDM-Miner, a novel framework for performing knowledge discovery from sparse and high-dimensional medical records. We have circumvented the drawbacks of traditional statistical analysis tools to develop a set of data-preprocessing, dimensionality reduction and finally knowledge discovery algorithms from medical data. We have proposed a detailed theoretical framework to characterize the dataset and provide a formal representation of the techniques that we used in this thesis. We evaluated SDM-Miner with experiments on synthetic as well as on real datasets, and showed the effectiveness of the overall framework in terms of low prediction error, low uncertainty in rule discovery, and high sensitivity and specificity of the algorithms. The experimental results clearly showed some promising results. Our proposed algorithm for estimating temporally missing values could handle the sparsity of medical records, and our dimensionality reduction algorithm derived an error-bounded compression of the dataset which aided in the generation of rules.

We plan to apply the dimensionality reduction algorithm to heterogeneous datasets which consist of a mix of categorical, discrete and continuous attributes. Reducing the error while compressing the dataset is also a future research goal. We have limited the knowledge discovery process to only two tasks - namely rule-mining and prediction. This is expected to be expanded to a range of other data mining tasks like pattern-mining and so on. Furthermore one major criticism of our work could be that it is too restrictive by focusing only on a restrictive dataset. We plan to test SDM-Miner to other datasets in the future and propose a more generic framework which would be applicable to all datasets showing the characteristics of the dataset that we have studied in this thesis. In particular, any data which falls under the category of being heterogeneous, sparse, high-dimensional and with security and privacy issues can be studied, and it would be interesting to analyse the performance of SDM-Miner as applied to this type of data.

References

[1] Y. Yin, B. Zhang, Y. Zhao, and G. Wang. Mining the most interesting patterns from multiple phenotypes medical data. In *RSCTC*, pages 696–705, 2006.

[2] S. Ghazavi and T. Liao. Medical data mining by fuzzy modeling with selected features. *Artificial intelligence in medicine*, pages 195–206, 2008.

[3] B. Robson. Clinical and pharmacologic data mining; generalized theory of expected information and application to the development of tools. In *J. Proteom. Res.*, pages 283–302, 2003.

[4] K. Cios and G. Moore. Uniqueness of medical data mining. In *Artificial Intelligence in Medicine*, pages 1–24, 2002.

[5] R. Bharat Rao, R. Rosales, S. Niculescu, S. Krishnan, L. Bogoni, X.S. Zhou, and B. Krishnapuram. Mining medical records for computer aided diagnosis. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[6] E. Ryan, B. Paty, P. Senior, D. Bigam, E. Alfadhli, N. Kneteman, J. Lakey, and A. Shapiro. Five-year follow-up after clinical islet transplantation. In *Diabetes*, pages 2060–2069, 2005.

[7] E. Rafael, E.A. Ryan, B. Paty, J. Oberholzer, S. Imes, P. Senior, C. McDonald, J. Lakey, and A. Shapiro. Changes in liver enzymes after clinical islet transplantation. In *Transplantation*, pages 1280–1284, 2003.

[8] M.A. Piper, J. Seidenfeld, N. Aronson. Islet Transplantation in Type 1 Diabetes Mellitus. Evidence Report/Technology Assessment No. 98 (Prepared by the Blue Cross and Blue Shield Association Technology Evaluation Center Evidence-based Practice Center under Contract No. 290-02-0026). *AHRQ Publication No. 04-E017-2*, Rockville, MD: Agency for Healthcare Research and Quality, 2004. www.ahrq.gov/downloads/pub/evidence/pdf/islet/islet.pdf

[9] C.C. Aggarwal. Towards Effective and Interpretable Data Mining by Visual Interaction. *SIGKDD Explorations* 3(2): pages 11–22, 2002.

[10] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer-Verlag*, 2001

[11] A. Sami. Obstacles and Misunderstandings Facing Medical Data Mining. *ADMA*, pages 856–863, 2006.

[12] J. Han and M. Kamber. Data mining: concepts and techniques. In *Morgan Kauffman*, 2001.

[13] S. Wang. Nonlinear pattern hypothesis generation for data mining. *Data and Knowledge Engineering*, Volume 40, Issue 3, pages 273–283, 2002.

[14] D. Delen, G. Walker, and A. Kadam. Predicting breast cancer survivability: a comparison of three data mining methods. In *Artificial Intelligence in Medicine*, pages 113–127, 2004.

[15] D. Collet. Modeling Survival Data in Medical Research. ed. C. Chatfield, J.V. Zidek, London: *Chapman and Hall*, 1994.

[16] B. H.B., D. Rosen, and P. Goodman. Comparing the prediction accuracy of artificial neural networks and other statistical models for breast cancer survival. In *Advances in Neural Information Processing Systems*, pages 1063–1067, 1995.

[17] G. Santos-Garcia, G. Varela, N. Novoa, and M. Jimenez. Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble. In *Artificial Intelligence in Medicine*, pages 61–69, 2004.

[18] I. Mullins, M. Siadaty, J. Lyman, K. Scully, C. Garrett, W. Miller, R. Muller, B. Robson, C. Apte, S. Weiss, I. Rigoutsos, D. Platt, S. Cohen, and W. Knaus. Data mining and clinical data repositories: Insights from a 667,000 patient data set. In *Comput Biol Med*, pages 1351–1377, 2006.

[19] E.D. Peterson, L.P. Coombs, E.R. DeLong, C.K. Haan, T.B. Ferguson. Procedural volume as a marker of quality for CABG surgery. In *J. Am. Med. Assoc.*, pages 195201, 2004.

[20] A. Abu-Hanna and de Keizer N. Integrating classification trees with local logistic regression in intensive care prognosis. In *Artif. Intell. Med.*, pages 5–23, 2003.

[21] R. Korrapati, S. Mukherjee, and K. Chalam. A bayesian framework to determine patient compliance in glaucoma cases. In *Proc. AMIA Symp.*, page 1050, 2000.

[22] E. Brossette, A. Sprague, J. Hardin, K. Waites, W. Jones, and S. Moser. Association rules and data mining in hospital infection control and public health surveillance. In *Journal of American Medical Informatics Association*, pages 373—381, 1998.

[23] M. Ohsaki, Y. Sato, H. Yokoi, and T. Yamaguchi. A rule discovery support system for sequential medical data in the case study of a chronic hepatitis dataset. In *ECML/PKDD-2003 Discovery Challenge Workshop*, pages 154—165, 2003.

[24] J. Harrison. Introduction to the mining of clinical data. In *Clinics in Laboratory Medicine*, pages 1–7, 2008.

[25] N. Lavrac, M. Bohanec, A. Pur, B. Cestnik, M. Debeljak, and A. Kobler. Data mining and visualization for decision support and modeling of public health-care resources. In *Journal of Biomedical Informatics*, pages 438–447, 2007.

[26] R. Hylock, W.N. Street, D.F. Lu, F. Currim. NursingCareWare: Warehousing and knowledge discovery for a nursing care data set. In *INFORMS Workshop on Data Mining and Health Informatics*, 2008.

[27] L. Durand, M. Blanchard, G. Cloutier, H. Sabbah, P. Stein. Comparison of pattern recognition methods for computer-assisted classification of spectra of heart sounds in patients with a porcine bioprosthetic valve implanted in the mitral position. In *IEEE Transactions on Biomedical Engineering* pages 11211129, 1990.

[28] X. Wang, M.R. Smith, R.M. Rangayyan. Mammographic information analysis through association-rule mining. In *IEEE Canadian Conference on Electrical and Computer Engineering*, pages 1495-1498, 2004.

[29] H. Alto, R.M. Rangayyan, J.E.L. Desautels. Content-based retrieval and analysis of mammographic masses. In *Journal of Electronic Imaging*, 2005 (Article 023016, pages 1–17. Erratum: 16(1), 019801:1 (JanMar) 2007).

[30] J.F. Roddick, P. Fule, W.J. Graco. Exploratory medical knowledge discovery: experiences and issues. *SIGKDD Explorations* pages 94–99, 2003.

[31] M. Hauskrecht, M. Valko, B. Kveton, S. Visweswaram, G. Cooper. Evidence-based anomaly detection in clinical domains. In *Annual American Medical Informatics Association (AMIA) conference*,pages 319–323, 2007.

[32] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *VLDB*, pages 420–431, 1995.

[33] D.A. Penn Estimating Missing Values from the General Social Survey: An Application of Multiple Imputation. In *Social Science Quarterly*, pages 573–584, 2007.

[34] D.B. Rubin. Multiple Imputation for Nonresponse in Surveys. *John Wiley and Sons*, 1987.

[35] N. Metropolis, S. Ulam. The Monte Carlo Method. In *Journal of the American Statistical Association*, pages 335–341, 1949.

[36] M.S.B. Sehgal, I. Gondal and L.S. Dooley. Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data. In *Bioinformatics*, pages 2417–2423, 2005.

[37] I.K. Fodor. A survey of dimension reduction techniques. *LLNL technical report*, UCRL-ID-148494, 2002.

[38] B. Liu, W. Hsu, Y. Ma. Integrating Classification and Association Rule Mining. In *KDD*, pages 80–86, 1998.

[39] R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB*, pages 487–499, 1994.

[40] V. Jovanoski and N. Lavrac. Classification Rule Learning with APRIORI-C. In *EPIA*, pages 44–51, 2001.

[41] J.M. Bernardo. Model-Free objective Bayesian prediction. In *Rev. Real Academia Ciencias Madrid*, pages 295–302, 1999.

[42] J.K. Jones. The role of data mining technology in the identification of signals of possible adverse drug reactions: value and limitations. In *Current Therapeutic Research*, pages 664–672, 2001.

[43] M.R. Kraft, K.C. Desouza and I. Androwich. Data Mining in Healthcare Information Systems: Case Study of a Veterans? Administration Spinal Cord Injury Population. In *Hawaii International Conference on System Sciences*, vol. 6, pages 159a, 2003.

[44] H. Khanna Nehemiah, A. Kannan, K. Vijaya, Y. Nancy Jane and J. Brindha Merin. Employing Clinical Data Sets for Intelligent Temporal Rule Mining and Decision Making, A Comparative Study. In *ICGST International Journal on Bioinformatics and Medical Engineering, BIME*, pages 37–45, 2007.

[45] I.R. John and P.R. Innocent. Modeling Uncertainty in Clinical Diagnosis Using Fuzzy Logic. *IEEE transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, pages 1340–1350, 2005.

[46] K. Polat, S. Sahan and S. Gunes. A new method to medical diagnosis: Artificial immune recognition system (AIRS) with fuzzy weighted pre-processing and application to ECG arrhythmia. In *Expert Systems with Applications*, pages 264–269, 2006.

[47] M. Koyutrk and A. Grama. PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. In *KDD*, 147-156, 2003.

[48] T.G. Kolda and D.P. O'Leary. Algorithm 805: computation and uses of the semidiscrete matrix decomposition. In *ACM Trans. Math. Softw.*, pages 415–435, 2000.

[49] G. Karypis and V. Kumar A fast and high quality multilevel scheme for partitioning irregular graphs. In *SIAM Journal on Scientific Computing*,pages 359–392, 1998.

[50] S. Schmidt, P. Vuillermin, B. Jenner, Y. Ren, G. Li and Yi-P.P. Chen Mining Medical Data: Bridging the Knowledge Divide. In *eResearch Australasia 2008.*

[51] S. Schmidt, G. Li, Yi-P.P. Chen. Medical Knowledge Discovery from a Regional Asthma Dataset. In *The 4th International Conference on Intelligent Computing (ICIC)*, pages 888–895, 2008.

[52] Australian Institute of Health and Welfare. Chronic respiratory diseases in Australia Their prevalence, con- sequences and prevention. AIHW Cat. No. PHE 63. Canberra: AIHW, 2005.

[53] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara and S. Ishii. A bayesian missing value estimation method for gene expression profile data. In *Bioinformatics*, pages 2088–2096, 2003.

[54] P. Liu, E. El-Darzi, L. Lei, C. Vasilakis,P. Chountas and W. Huang. Applying data mining algorithms to inpatient dataset with missing values. In *Journal of Enterprise Information Management*, pages 81–92, 2008.

[55] C.L. Lawson and R.J. Hanson. Solving Least Squares Problems. In *Englewood Cliffs, NJ Prentice-Hall*, 1974.

[56] S. Timotheou. Nonnegative Least Squares Learning for the Random Neural Network. In *ICANN*, pages 195–204, 2008.

[57] K.V. Mardia, J.T. Kent, and J.M. Bibby. Multivariate Analysis. In *Probability and Mathematical Statistics*, Academic Press, 1995.

[58] I.T. Jolliffe. Principal Component Analysis. In *Series: Springer Series in Statistics*, 2nd ed., Springer, NY, 2002, XXIX, 487.

[59] J. H. Friedman and J. W. Tukey. A Projection Pursuit Algorithm for Exploratory Data Analysis. In *IEEE Transactions on Computers*, pages 881–890, 1974.

[60] J. Karhunen, P. Pajunen, and E. Oja. The nonlinear pca criterion in blind source separation: Relations with other approaches. In *Neurocomputing*, pages 5–20, 1998.

[61] A. Hyvarinen, J. Karhunen, and E. Oja. Independent Component Analysis. In *Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control*, Wiley, 2001.

[62] R. Malik, S. Kim, X. Jin, C. Ramachandran, J. Han, I. Gupta, K. Nahrstedt. MLR-Index: An Index Structure for Fast and Scalable Similarity Search in High Dimensions. In *SSDBM*, pages 167–184, 2009.

[63] V. Jovanoski and N. Lavrac. Classification Rule Learning with APRIORI-C. In *EPIA*, pages 44-51, 2001.

[64] R. B. OHara, E. Arjas, H. ,Toivonen and I. Hanskii Bayesian Analysis Of Metapopulation Data. In *Ecology*, pages 2408–2415, 2002.